MAKİNE ÖĞRENME YÖNTEMLERİ VE KELİME KÜMESİ TEKNİĞİ İLE İSTENMEYEN E-POSTA / E-POSTA SINIFLAMASI

SPAM / HAM E-MAIL CLASSIFICATION USING MACHINE LEARNING METHODS BASED ON BAG OF WORDS (BOW) TECHNIQUE

ESRA ŞAHİN

YRD. DOÇ. DR. MURAT AYDOS

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2018

ESRA ŞAHİN' in hazırladığı **"Makine Öğrenme Yöntemleri Ve Kelime Kümesi Tekniği İle İstenmeyen E-Posta/E-Posta Sınıflaması"** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'**nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Mehmet Önder EFE
Başkan ..... 

Yrd. Doç. Dr. Murat AYDOS
Danışman ..... 

Yrd. Doç. Dr. Adnan ÖZSOY
Üye ..... 

Doç. Dr. Ahmet Burak CAN
Üye ..... 

Doç. Dr. Suat ÖZDEMİR
Üye ..... 

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

☐ **Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**
(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etseniz ve kütüphane bu talebinizi yerine getirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

☒ **Tezimin/Raporumun 30.04.2018 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**
(Bu sürenin sonunda uzatma için başvuruda bulunmadığım taktirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

☐ **Tezimin/Raporumun ..... tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

☐ **Serbest Seçenek/Yazarın Seçimi**

18 / 03 /2018

Esra ŞAHİN

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

22.02.2018

ESRA ŞAHİN

ÖZET

MAKİNE ÖĞRENME YÖNTEMLERİ VE KELİME KÜMESİ TEKNİĞİ İLE İSTENMEYEN E-POSTA / E-POSTA SINIFLAMASI

Esra ŞAHİN

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Murat AYDOS

Şubat 2018, 60 Sayfa

Günümüzde elektronik ortamda sıkça kullandığımız iletişim yollarından birisi olan e-postalar; kişisel iletişimler, iş odaklı aktiviteler, pazarlama, reklam, eğitim vs. gibi birçok nedenden dolayı hayatımızda önemli bir yer kaplamaktadır. Birçok farklı konudaki iletişim ihtiyacı için hayatımızı kolaylaştıran e-postalar, kötü niyetli göndericiler tarafından kullanıldıklarında ise alıcıların zamanını çalması, maddi ve manevi kayıplara sebep olması nedeniyle hayatı oldukça zorlaştırabilmektedir. Tanımadığımız ya da güvenilmeyen adreslerden, reklam ya da güvenlik tehdidi oluşturma amaçlı gönderilen e-postalar, bilgi güvenliği açısından önemli tehlikeler oluşturabilir. Bu türdeki istenmeyen e-postaları, insanlara zarar vermeden önce tespit edip önleyebilmek ise ayrı bir çalışma konusu olmuştur.

Bu tez çalışmasında istenmeyen e-postalar kapsamlı bir şekilde araştırılmış, istenmeyen e-postaları sınıflandırmak için yapılan çalışmalar incelenmiştir. Alanyazındaki çalışmalardan farklı olarak e-posta içeriğinde yer alan linklerin metinleri ele alınarak, makine öğrenmesi yöntemleri ve Kelime Kümesi Tekniği ile istenmeyen e-posta/e-posta sınıflaması yapılmıştır. Yapılan çalışmada doğruluk, F1 skor ve sınıflama hata oranı metrikleri kullanılarak farklı

makine öğrenme tekniklerinin istenmeyen e-posta sınıflandırılmasındaki başarısı analiz edilmiştir. Diğer yandan başarı oranı %95 üzerinde çıkan makine öğrenme teknikleri için Kelime Kümesi Tekniği (BOW) ile elde edilen farklı N gramların sınıflandırma başarısına olan etkisi incelenmiştir. Çalışma sonucunda Bayes, Destek Vektör Makineleri, Sinir Ağları ve En Yakın Komşu algoritmaları yüksek başarı gösterirken Karar Ağaçları Algoritmalarının istenmeyen e-posta sınıflamada düşük başarı gösterdiği görülmüştür. Diğer yandan 5 gramların performansa en iyi katkıyı sağladığı görülmüştür.

Anahtar Kelimeler: Kelime Kümesi Tekniği, İstenmeyen E-posta, Makine Öğrenmesi Teknikleri

ABSTRACT

SPAM-HAM E-MAIL CLASSIFICATION USING MACHINE LEARNING METHODS BASED ON BAG OF WORDS (BOW) TECHNIQUE

Esra ŞAHİN

Master of Science, Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Murat AYDOS

February 2018, 60 Pages

Nowadays, we frequently use e-mails, which is one of the communication channels, in electronic environment. It plays an important role in our lives because of many reasons such as personal communications, business-focused activities, marketing, advertising, education, etc. E-mails make life easier because of meeting many different types of communication needs. On the other hand they can make life difficult when they are used outside of their purposes. Spam emails can be not only annoying receivers, but also dangerous for receiver's information security. Detecting and preventing spam e-mails has been a separate issue.

In this thesis, spam e-mails have been studied comprehensively and studies which is related to classifying spam e-mails have been investigated. Unlike the studies in the literature, in this study; the texts of the links placed in the e-mail body are handled and classified by the machine learning methods and the Bag of Words Technique. In this study, we analyzed the effect of different N grams on classification performance and the success of different machine learning techniques in classifying spam e-mail by using accuracy, F1 score and classification error

metrics. On the other hand, the effect of different N grams is examined for machine learning success rate of over %95. As a result of the study, it has been seen that Decision Trees Algorithms show low success in spam classification when Bayes, Support Vector Machines, Neural Networks and Nearest Neighbor Algorithms show high success. On the other hand, 5 grams were found to provide the best contribution for performance.

Keywords: Bag of Words Technique, Spam Email, Machine Learning

TEŞEKKÜR

Tez çalışmamın her aşamasında değerli katkılarıyla yol gösteren ve beni her zaman çalışmaya teşvik eden tez danışman hocam, Yrd. Doç. Dr. Murat AYDOS'a, önemli yorum ve yönlendirmeleriyle katkıda bulunan değerli jüri üyelerim Prof. Dr. Mehmet Önder EFE'ye, Yrd. Doç. Dr. Adnan ÖZSOY'a, Doç. Dr. Ahmet Burak CAN'a, Doç. Dr. Suat ÖZDEMİR'e teşekkür ederim.

Eğitim ve kariyer hayatım boyunca, desteklerini hep arkamda hissettiğim, hayatını ailesine adamış sevgili anneme ve babama, meslektaşım sevgili kardeşim Enes ŞAHİN'e minnettarım.

Çalışmalarımda beni her zaman teşvik eden, bana motivasyon katan sevgili dostum Arş. Gör. Begüm MUTLU olmak üzere ismini sayamadığım tüm dostlarıma teşekkür ederim.

Kariyer hayatıma büyük bir katkı sağlayan, iş hayatımı akademi ile desteklememe fırsat tanıyan ASELSAN A.Ş'ye teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Sayfa

ŞEKİLLER

Sayfa

KISALTMALAR

SPAM	İstenmeyen E-posta
SVM	Destek Vektör Makineleri (Support Vector Machine)
ARPANET	Gelişmiş Araştırma Projeleri Dairesi Ağı
MALWARE	Kötücül Amaçlı Yazılım (Malicious Software)
FTC	Federal Ticaret Komisyonu (Federal Trade Commission)
BOW	Kelime Kümesi Tekniği (Bag of Words)
KNN	En Yakın Komşu (K Nearest Neighborhood)
YSA	Yapay Sinir Ağları
İst. EP	İstenmeyen E-posta
USD	Amerikan Doları
TM	Turing Makinesi

1. GENEL BİLGİLER

Elektronik ortamda sıkça kullandığımız iletişim yollarından birisi olan e-postalar (e-mail); kişisel, iş odaklı, pazarlama, eğitim ve diğer türlü iletişim amaçları için milyonlarca insan tarafından sıkça kullanılmaktadır. Birçok farklı konudaki iletişim ihtiyacı için hayatımızı kolaylaştıran e-postalar, kötü niyetli göndericiler tarafından kullanıldıklarında ise alıcıların zamanını çalması, maddi ve manevi kayıplara sebep olması nedeniyle hayatı oldukça zorlaştırabilmektedir. Tanımadığımız ya da güvenilmeyen adreslerden, reklam ya da güvenlik tehdidi oluşturma amaçlı gönderilen e-postalar, "istenmeyen e-posta" *(spam, junk e-mail)* olarak adlandırılmaktadır [1].

Özellikle pazarlama ve reklam yöntemi olarak çok sayıda işletme potansiyel müşterilerinin mail adreslerini toplamayı denemektedir. Bu işletmelerin bazıları kötü niyetli davranarak topladığı bu bilgileri diğer ticari işletmelerle paylaşmaktadır [1]. Bunun sonucunda ise binlerce insan istenmeyen reklam içerikli maillerle karşı karşıya kalmaktadır. Stephen tarafından yapılan çalışmada istenmeyen e-posta ekonomisinin gerçek hayatta şu örnekle anlatılmıştır [2]; Perakende fiyatı 49.95 $ olan yeni bir bilgisayar programı satmak istediğinizi varsayalım. Yaptıkları her satış için bir pazarlama şirketine 19 $ ödenecektir. Geleneksel reklam yaklaşımı ile potansiyel bir müşteriye gönderdiğiniz her broşür, en az 1.00 USD oluşturmaktadır. 5.000 broşürü postaladığımızı düşünürsek sadece posta maliyetini karşılaması için % 5'ten daha iyi bir geri dönüş oranına ihtiyacınız var (5,000 USD ödemeniz için her biri 19 USD komisyon ile 263 birim satmanız gerekiyor). Toplu e-posta yolu ile reklam yapmak ise durumu oldukça farklı bir hale getirir. Wall Street Journal'ın (11/13/02) yaptığı bir araştırma, e-postaları kullanırken % 0.001 gibi düşük bir getiri oranının karlı olabileceğini göstermektedir. 5 milyon mesajın gönderildiği bu örnekte, ilk haftada %0.0023 oranında 81 satışla sonuçlanmıştır. Her satışın pazarlama şirketine 19 USD değerinde geri dönüş yaptığı düşünülünce toplamda 1.500 USD gelir elde edilmiştir. Buradaki 5000 iletiyi gönderme maliyeti çok düşüktür. 56 Kbps modem üzerinde bile saatte binlerce mesaj gönderebilir. Bu durum çalışmasında da görüldüğü gibi istenmeyen e-postalar, düşük maliyetle reklam ve pazarlama ile yüksek kazançlar sağlamaktadır.

Diğer yandan istenmeyen e-postalar bilgi güvenliği açısından da önemli tehlikeler oluşturabilir. Bu tehlikeyi insanlara zarar vermeden önce tespit edip önleyebilmek ise ayrı bir çalışma konusu olmuştur. Bilgisayar ağları üzerindeki tehlikeler çeşitlendikçe, büyük ölçekli çevrimiçi servis sağlayıcıları, kullanıcıları oluşabilecek saldırılardan korumak için

savunma sistemlerini geliştirmektedir. Saldırganlar ise, çabalarını platform üzerindeki doğrudan saldırılardan sosyal mühendislik saldırılarına kaydırmaya başlamışlardır [3]. Diğer bir deyişle; saldırganlar, kullanıcılara saldırabilmek için kullanıcılar tarafından yapılan yanlış eylemlerden yararlanmaya çalışmaktadırlar. Örneğin bir hesabın bilgilerini ele geçirmek amacıyla saldırı yapanlar kullanıcının saldırıyı algılamasını engellemek amacıyla istenmeyen e-posta göndermeyi denemektedirler [3].

İstenmeyen e-postaların geçmişine baktığımızda ise ilk istenmeyen e- postanın 1978 yılında ARPANET üzerinden gönderildiğini görmekteyiz [5]. İnternetin dünyamıza girmesinden kısa bir süre sonra hayatımıza giren istenmeyen e-postalarla mücadele geçmişten günümüze kadar sürmüştür. Bu durum istenmeyen e-postalardan korunma yöntemlerinin sürekli geliştiğini ve değiştiğini ancak istenmeyen e-postaların tamamen engellenemediğini, bu alanda yapılacak çalışmalara günümüz de de ihtiyaç olduğunu göstermektedir.

2017 yılı İnternet Güvenliği Tehdit Raporu'na bakıldığında e-postaların yarısından fazlasının (%53) istenmeyen e-posta, bu e-postaların artan bir kısmının da kötücül amaçlı yazılımlar (malware) içerdiği görülmektedir [4]. 2015 yılında 220 e-postadan 1 tanesinin kötücül amaçlı yazılım içerdiği görülürken, 2016 yılında bu oran 131 e-posta da 1'e yükselmiştir [4]. E-posta ile bulaşan kötücül amaçlı yazılımlardaki bu artış, bu tür yazılımların büyük oranda profesyonelleştiğini, bir diğer yandan ise saldırganların bu tür saldırılardan önemli kazançlar sağladıklarını ve bu uygulamaya devam etmelerinin muhtemel olduğunu göstermektedir [4]. Bu yazılımların değişen içeriklerine bakıldığında, önceleri büyük çaplı tehlikeli e-posta grupları kullanıcıların bir şeyler indirmesini ve yüklemesini, daha sonra ise bir miktar ödeme diğer bir deyişle fidye karşılığında kitlenen bilgisayar sistemlerinin yeniden açılmasını sağlamaktaydılar. 2016 yılının başlarına bakıldığında ise istenmeyen e-posta şirketlerinin en çok kötücül amaçlı makrolar içeren Office dokümanlarını kullandıkları görülmüştür. Bu konuda kötü amaçlı yazılımların başlıca dağıtıcılarından biri olarak bilinen bir Necurs'un, JavaScript ve Office makro ekleri aracılığıyla kötü amaçlı yazılımları yaymasının etkisi büyüktür [4]. Yapılan saldırıların şirketlere olan etkilerine bakıldığında küçük çaplı şirketlere gönderilen istenmeyen-eposta sayılarının, büyük çaplı şirketlere gönderilen istenmeyen e-postalara göre oldukça düşük olduğu görülmektedir. Bu durum saldırıların büyük çaplı kazanç sağlayacakları şirketlere yönelmesinin muhtemel olduğunu göstermektedir.

İstenmeyen e-postaları tanımlamak istediğimizde aşağıdaki üç kriterden birine uyan herhangi bir e-postadır diyebiliriz. Bu kriterler aşağıda verilmiştir [6]:

* İsimsizlik (Anonymity): Gönderenin adresi ve kimliği gizlidir.

* Toplu posta: E-posta toplu gruplara reklam, pazarlama gibi amaçlar ile gönderilmiştir.

* İstenmeyen (Unsolicited) : E-postalar alıcı tarafından istenmemektedir.

Birçok insan gelen kutusundaki istenmeyen (spam) bir e-postayı kolayca ayırt edebilir ve yok sayabilir. Çoğu zaman kişiler spam e-postalarla ile ilgili tehlikelerde bilgili olmaması, gelen spam e-posta sayısının az olması gibi nedenlerle bu e-postalar rahatsız edici olmayabilir. Ancak gelen tek bir e-posta olsa dahi bu durum spam e-postaların yaratacağı tehlikeyi yok etmez. Tek bir spam mesajı yanıtlayarak spam göndericilerin istediği maddi veya kişisel bilgileri vermek mümkündür. Sonuç, spam gönderenler için kazanç, alıcılar için ise maddi kayıp olabilir.

Maliyet açısından bakıldığında ise istenmeyen e-postalara tepki verecek küçük bir kitle için bu kadar fazla kişiye e-posta göndermek mantıklı olup olmadığı akla gelmektedir. Spam gönderen kişi ne kadar çok kişiye ulaşırsa, alıcıların e-postaya cevap verme olasılığı da o kadar yüksektir. Diğer yandan bu e-postaları göndermenin maliyeti, spam göndericilerin kendi sunucularını kullanması veya düşük maliyetli Proxy sunucular kiralamasından dolayı oldukça düşüktür [6].

Yasalar açısından bakıldığında ise ABD Federal Hükümeti, özellikle Federal Ticaret Komisyonu (FTC), istenmeyen e-posta konusunu gündeme almış ve CAN-SPAM federal mevzuatını 2003 yılında yürürlüğe koymuştur. FTC'nin görevi, tüketici haklarını korumaktır. İstenmeyen e-postalar tüketicileri iki farklı şekilde riske atabilir [6]:

- Mali ve Gizlilik Riskleri: Hedeflenen durum alıcının finansal bilgilerine ulaşmak olduğu için çoğu istenmeyen e-postalar alıcıdan kredi kartı numaraları veya sosyal güvenlik numarası gibi kişisel verileri talep eder. Bu veriler daha sonra kimlik hırsızlığı, kredi kartı sahtekârlığı gibi işler için kullanılabilir. CAN-SPAM'a göre yakalanan herhangi bir istenmeyen e-posta gönderene bu nedenle ceza verilebilir [6].
- Çocukları Koruma: Bir e-posta göndericisinin e-postayla gönderilen bir kullanıcının yaşını bilmesinin imkânı yoktur. CAN-SPAM, çocuklara uygun olmayan istenmeyen e-postaları ortadan kaldırmak amacıyla geliştirilmiştir [6].

Kısacası CAN-SPAM, e-posta hesabınıza istenmeyen e-posta akışı durdurma da bir filtre gibi görev yapmamış olsa da çoğu e-posta servis sağlayıcılarının, istenmeyen e-postaları gelen kutusundan uzak tutması için korku oluşturmaktadır. Aksi durumda cezalandırma işlemi yapılmaktadır. Gerçekte pazarlama amacıyla atılan e-postalar ile istenmeyen e-postaları ayırt etmek için CAN-SPAM şu kurallara dikkat edilmesi gerektiğini söyler [7];

- Söylediğiniz kişi olun: Bir kullanıcının bir e-postayı açmasını sağlamak ve ya e-postanın önemsiz hale gelmesini önlemek için başka bir web sitesi veya şirket gibi davranamazsınız. Bu, istenmeyen e-postaların filtrelere takılmasını engellemek için kullandıkları popüler bir hiledir, ancak yasa dışıdır.
- Konu satırında dürüst olun.
- Bir reklam olduğunuzu alıcılara açıkça anlatın.
- Gerçek fiziksel bir adres verin: Bu, bir dolandırıcı olmadığınızı ve müşterilere doğrulanmış bir iletişim yolunu göndermenizi sağlar.
- Alıcıların istek durumunda e-posta listesinden çıkabilmeleri için yol gösterin.
- Alıcıların istemesi durumunda onları e-posta listenizden çıkartın.

Bir e-posta servis sağlayıcılarının temel müşterisi e-posta hesabı olan kişilerdir. Çoğu e-posta hizmeti sağlayıcısı, bir kullanıcının gelen kutusunda harcadığı süreyi temel alarak gelirlerini artırır. Örneğin, çoğu çevrimiçi e-posta servis sağlayıcıları bir e-posta gelen kutusunun çevrimiçi sürümü içinde web tabanlı reklamlar sunar. Kullanıcıların e-posta kutusunda geçireceği zaman arttıkça servis sağlayıcının sunduğu reklama tıklama olasılığı artar, daha fazla reklam gösterme şansı doğar. İstenmeyen e-postalarla dolu bir gelen kutusunda zaman kaybetmek yerine kullanıcı, gerçekten önemli maillerine ulaşmakta daha az zaman harcayacağı ve daha iyi deneyimlerle karşılaşacağı başka bir e-posta hizmeti sağlayıcısına geçmeyi deneyebilir. Bu nedenle e-posta hizmet sağlayıcıları iyi istenmeyen e-posta filtrelerine sahip olmaya çalışırlar.

1.1 Tezin Amaç ve Hedefleri

İstenmeyen e-postaları sınıflandırmak için yapılan çalışmalar incelendiğinde çok sayıda teknik ile e-postaların başlık (header), içerik (body) bölümlerindeki çeşitli bilgiler kullanılarak sınıflandırma yapıldığı görülmüştür. Bu çalışmalardan farklı olarak bu tez çalışmasının temel amacı e-postaların içeriklerinde yer alan bağlantıların metinlerine odaklanmasıdır. Çalışmada veri setlerinin oluşturulması ve vektörel gösterimi aşamasında Kelime Kümesi Tekniği, sınıflandırma işlemleri için ise makine öğrenmesi teknikleri

kullanılmıştır. BOW sonucu oluşan farklı uzunluktaki N gramların sınıflandırma performansına etkisi ve farklı makine öğrenme tekniklerinin istenmeyen e-posta sınıflandırması için başarısı analiz edilmiştir.

Bu doğrultuda tez çalışmasının amaç ve hedefleri aşağıdaki gibidir;

- İstenmeyen e-postalar, istenmeyen e-postaların oluşturacağı tehlikeler, maliyetler ve istenmeyen e-postaları gönderirken hedeflenen temel amaçlar hakkında bilgi verilmesi
- İstenmeyen e-postaları ayırt etmek için yapılan çalışmalar hakkında bilgi verilmesi, bu çalışmalarda yer alan yöntemlerin ve modellerinin incelenmesi.
- İstenmeyen e-postaları alanyazında yer alan çalışmalardan farklı olarak bağlantı metinlerine göre sınıflandırma yönteminin belirlenmesi
- İstenmeyen e-postaların özelliklerinin çıkartılması aşamasında kullanılan Kelime Kümesi Tekniği ve bu tekniğin uygulanması amacıyla tasarlanan algoritma hakkında bilgi verilmesi
- Farklı makine öğrenmesi tekniklerinin istenmeyen e-postaları sınıflandırmadaki başarısının gözlemlenmesi
- Sınıflandırma başarısını ölçen metriklerinin incelenmesi
- Bağlantı metinlerine göre sınıflama yönteminin gerçeklenerek test edilmesi
- Önerilen modelin mevcut çözümlerle karşılaştırılması.

1.2 Tezin Kapsamı

Tez kapsamında; istenmeyen e-postaları, Kelime Kümesi Tekniği ve farklı makine öğrenmesi yöntemleri kullanılarak sınıflama yöntemi incelenmiştir. Seçilen makine öğrenmesi yöntemlerinin uygulanırken alanyazındaki çalışmalardan farklı olarak e-postalar içerisinde yer alan bağlantıların metinleri kullanılmıştır. Sınıflandırma başarısını ölçen metrikler incelenerek, oluşturulan N gramların sınıflandırma başarısına etkisi, farklı makine öğrenmesi tekniklerinin istenmeyen e-postaları sınıflandırmadaki başarısı karşılaştırılmıştır. Çalışma, içerisinde bağlantı barındıran istenmeyen e-postaların ayırt edilmesi için çözüm sunmaktadır.

1.3 Tezin Yapısı

Çalışmanın yapısı aşağıda maddeler halinde verilmiştir.

- Giriş bölümünde; istenmeyen e-postalar ana hatlarıyla özetlenmiş, çalışmanın amacı, hedefleri, kapsamı ve tezin yapısı hakkında bilgiler verilmiştir.

- İkinci bölümde; bu teze konu olan makine öğrenmesi yöntemleri hakkında bilgi verilmiştir.
- Üçüncü bölümde; istenmeyen e-postaların sınıflandırılması için alanyazına yerleşen kavramlar, modeller, kullanılan araçlar ve metriklerden bahsedilmiştir.
- Dördüncü bölümde; çalışmanın yöntemi tanıtılmıştır.
- Beşinci bölümde; çözüme yönelik önerilen modelin gerçekleştirimi yapılmış ve elde edilen sonuçlar ortaya konmuş ve tartışılmıştır.
- Altıncı bölümde, çalışmanın sonuçları sunulmuştur.

2. ALAN BİLGİSİ

Makine Öğrenmesi, bir probleme ait verilerden yola çıkarak modelleme işlemi yapan algoritmaların genel adıdır. Genel olarak "Danışmanlı (Supervised), Danışmansız (Unsupervised) ve Yarı Danışmanlı (Semi-Supervised) öğrenme olarak 3'e ayrılır [8].

Danışmanlı Öğrenme: Bir X girdi değeri ve bir Y çıktı değerinden yola çıkarak girdiden çıktıya eşleme fonksiyonunu öğrenen algoritmalardır. Diğer bir deyişle her bir verinin hangi sınıfa ait olduğunun bilindiği durumlarda kullanılan algoritmalardır [8]. Eşitlik 1'de X girdi, Y çıktı olmak üzere f eşleme fonksiyonu gösterilmektedir.

$$Y=f(X) \quad (1)$$

Eşitlik (1)'deki f eşleşme fonksiyonunun temel amacı yeni bir X girdisi geldiğinde Y çıktısını tahmin edebilmektir. Eğitim veri setinde tüm verilerin hangi sınıfa ait olduğu bilindiğinden ve bu verilerin öğrenme sürecinde rehberlik etmesinden dolayı "Danışmanlı Öğrenme" adı verilmiştir. Öğrenme işlemi, kabul edilebilir bir performans seviyesine ulaşıldığında durur. Danışmanlı Öğrenme; sınıflama ve regresyon olarak iki gruba ayrılabilir [8]. Sınıflandırma, çıktı değerinin kategorik olduğu durumlardır. Örneğin istenmeyen e-posta/ e-posta sınıflandırılması. Regresyon ise çıktı değerinin reel bir sayı olduğu durumlardır.

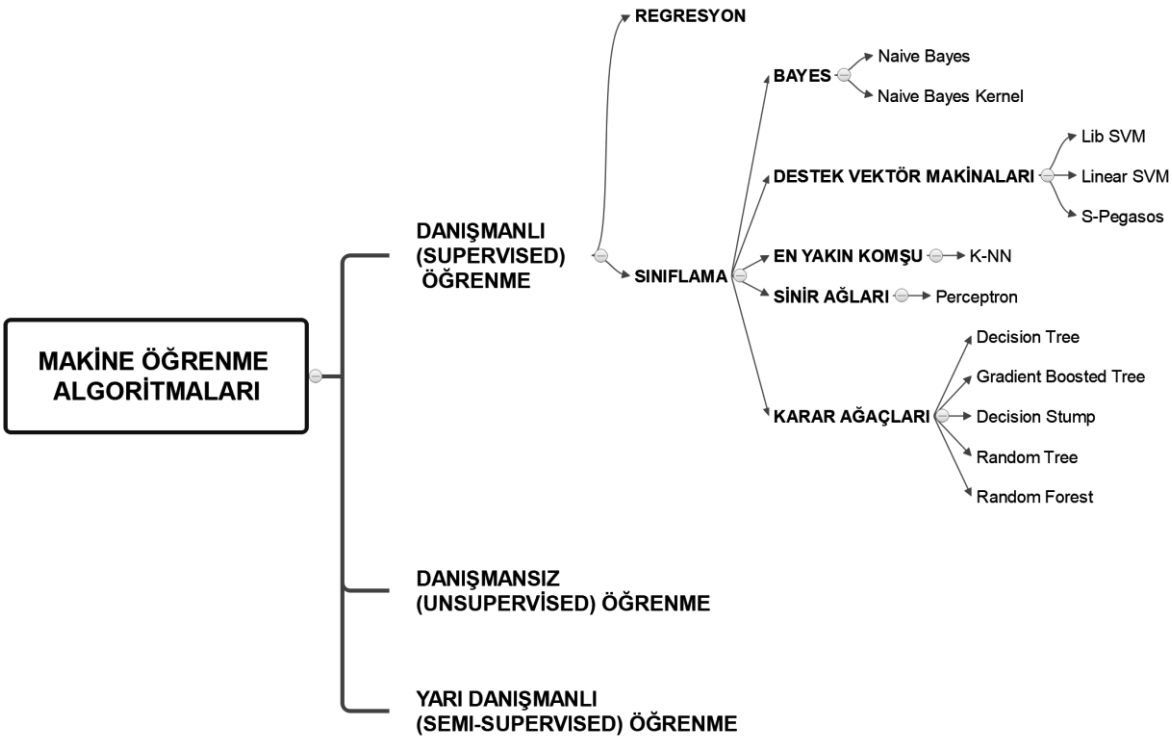
Danışmansız Öğrenme: Yalnızca X girdisinin olduğu ve bu X değerine karşılık gelen Y çıktısının bilinmediği durumlardaki öğrenme yöntemidir. Denetimsiz öğrenmenin amacı, veriler hakkında daha fazla bilgi edinmek için verinin altında yatan yapıyı veya dağılımı modellemektir. Danışmanlı Öğrenme'nin aksine burada öğrenme sürecine rehberlik edecek bir veri yoktur. Danışmansız Öğrenme; Kümeleme (Clustering) ve Birliktelik (Association)

olmak üzere ikiye ayrılır [8]. Kümeleme problemi, ayrık grupların keşfedildiği yerdir. Birliktelik ise verinin geniş bir bölümünü tanımlayan kuralları keşfetme problemidir [8].

Yarı Danışmanlı Öğrenme: X girdileri ve Y çıktılarından sadece bazı çıktıların etiketinin bilindiği öğrenme yöntemidir. Danışmanlı ve Danışmansız Öğrenme yöntemlerinin her ikisini de kapsar. Bir veriye etiket vermenin ayrı bir maliyeti vardır. Etiketsiz veriler ise bu anlamda daha ucuz, toplanması ve saklanması ise daha kolaydır. Bu açılardan pratik olması nedeniyle çoğu makine öğrenmesi yöntemi bu kapsama girer.

Girdi yapısını keşfetmek ve öğrenmek istediğimiz durumlarda Danışmansız Öğrenme, etiketlenmemiş veriler hakkında en iyi tahminleri yapmak istediğimiz durumlarda Danışmanlı Öğrenme yöntemleri kullanılmalıdır [8].

Tez çalışması kapsamında kullanılan verilerin hepsinin hangi sınıfa ait olduğu bilinmektedir. Bu nedenle yukarıda anlatılan Makine Öğrenmesi yöntemlerinden Danışmanlı Öğrenme algoritmalarından "sınıflama algoritmaları" tez kapsamında ele alınmıştır.



Şekil 2-1 Makine öğrenmesi yöntemleri

Tez kapsamında çalışılan Danışmanlı Öğrenme algoritmaları seçilirken temelde sınıflandırma başarısı göz önüne alınmıştır. Buna göre öncelikle daha sonraki bölümlerde açıklanan 50 limiti ile sınırlandırılmış 3 gramlık veri seti ile eğitim yapılmıştır. Bu veri setinin seçilmesinin nedeni özellik sayının diğer veri setlerine göre daha az olması ve gram

olarak ortalama bir değerde olmasıdır. Bu eğitimin sonunda başarısı %95'ının üzerinde olan algoritmalar için N gramlara göre performans değişimi incelenmiştir. %95 başarının altında kalan algoritmalarla ise sadece tek bir deney gerçekleştirilmiştir. N gramların performansa etkisinin incelendiği diğer bir deyişle %95 üzerinde doğruluk gösteren algoritmalar aşağıda detaylıca açıklanmıştır.

2.1 Bayes Sınıflandırıcı

2.1.1 Naive Bayes Sınıflandırıcı

Naive Bayes sınıflandırıcı, adını 17. yüzyılda yaşamış olan İngiliz matematikçi Thomas Bayes'ten alır [9]. Temelinde, bağımsız değişken varsayımlarını konu alan Bayes Teoremi yatar. Özellikle girdilerin boyutu yüksek olduğunda Naive Bayes uygun bir sınıflandırıcı olarak belirlenebilir. Basit bir mantık üzerine kurulmuş olmasına rağmen, Naive Bayes sınıflandırıcısı genelde şaşırtıcı derecede iyi sonuç verir, bu nedenle yaygın olarak kullanılır. Tipik kullanım alanları, gerçek zamanlı tahminleme, istenmeyen e-posta algılama, duyarlılık analizi, metin sınıflama vs. [10]. Bayes Teoremi Eşitlik (2)'deki gibidir [8];

$$P(A\backslash B) = \frac{P(B\backslash A) * P(A)}{P(B)} \quad (2)$$

P(A\B): B olayı gerçekleştiği anda A olayının meydana gelme olasılığı

P(B\A): A olayı gerçekleştiği anda B olayının meydana gelme olasılığı

P(A) ve P(B): A ve B olaylarının önsel olasılıklarıdır.

Naive Bayes ile sınıflandırma işlemi yaparken [8];

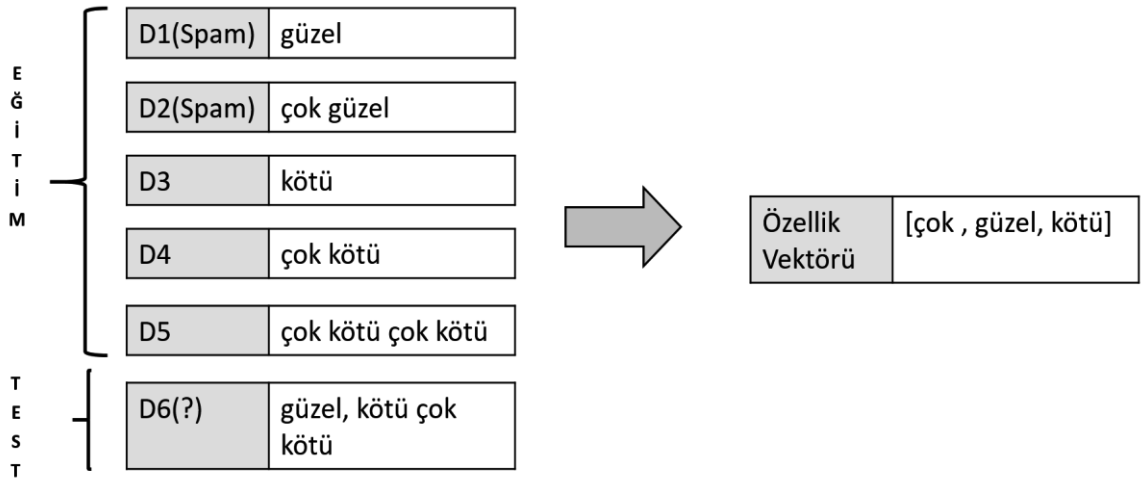
1. Veri seti frekans tablosuna dönüştürülür.
2. Olasılık sınıflandırılması için her bir değişkenin olasılığı hesaplanır.
3. Her sınıfın olasılığını hesaplamak için Naive Bayes denklemi kullanılır. En yüksek olasılıklı sınıf tahmin sonucudur.

Naive Bayes Sınıflandırıcı istenmeyen e-posta sınıflandırmasında sıkça tercih edilen yöntemlerden birisidir. İstenmeyen mail sınıflandırmasında bu teoremi aşağıdaki örnek ile açıklabilir [9];

- Her e-posta kelimelerden oluşan özellik vektörleri ile temsil edilir.
- X özelikleri göstermek üzere 30000 kelimelik bir özellik vektöründe aslında P(x|y) bulunmak istenmektedir. P(x|y) , Eşitlik (3)'deki gibi hesaplanır.

$$P(x_1, \ldots\ldots\ldots x_{3000}|y) \quad (3)$$
$$=P(x_1|y)P(x_2|y,x_1)P(x_3|y,x_1,x_2) \ldots\ldots P(x_{3000}|y,x_1,x_2 \ldots\ldots x_{2999})$$
$$=\prod_{i=1}^{n} P(x_i|y)$$

Örnek veri seti ve veri seti sonucu ortaya çıkan özellik vektörü Şekil 2.2'deki gibidir.



Şekil 2-2 Naive Bayes Sınıflandırma örnek veri seti

Özellik vektörü oluşturulduktan sonra Çizelge 2.1'deki gibi olasılıklar hesaplanır.

Çizelge 2-1 Naive Bayes- olasık hesaplama örneği

P(X)=X/(X+Y) (X: İstenmeyen E-Posta, Y: E-Posta)	2/(2+3)=0.4
P(Y) = Y/ (X+Y)=	3/(2+3)=0.6
P("Çok"/X)=	$N_{çok\|X}+1$)/{ ($N_{çok\|X}+1$)+ ($N_{güzel\|X}+1$)+ ($N_{kötü\|X}+1$)} =(1+1)/{(1+1)+(2+1)+(0+1)=0.33
P("Güzel"/X)=	($N_{güzel\|X}+1$)/{ ($N_{çok\|X}+1$)+ ($N_{güzel\|X}+1$)+ ($N_{kötü\|X}+1$)} =(2+1)/{(1+1)+(2+1)+(0+1)=0.5
P("Kötü"/X)=	($N_{kötü\|X}+1$)/{ ($N_{çok\|X}+1$)+ ($N_{güzel\|X}+1$)+ ($N_{kötü\|X}+1$)} =(0+1)/{(1+1)+(2+1)+(0+1)=0.17
P("Çok"/Y)=	$N_{çok\|Y}+1$)/{ ($N_{çok\|Y}+1$)+ ($N_{güzel\|Y}+1$)+ ($N_{kötü\|Y}+1$)}x =(3+1)/{(3+1)+(0+1)+(4+1)=0.4
P("Güzel"/Y)=	($N_{güzel\|Y}+1$)/{ ($N_{çok\|Y}+1$)+ ($N_{güzel\|Y}+1$)+ ($N_{kötü\|Y}+1$)} =(0+1)/{(3+1)+(0+1)+(4+1)=0.1
P("Kötü"/Y)=	($N_{kötü\|Y}+1$)/{ ($N_{çok\|Y}+1$)+ ($N_{güzel\|Y}+1$)+ ($N_{kötü\|Y}+1$)} =(4+1)/{(3+1)+(0+1)+(4+1)=0.5

Test verisi D6=”güzel, kötü çok kötü” için Bayes kuralı Çizelge 2.2’deki gibi hesaplanır. Hesaplama sonucunda D6 e-postasının istenmeyen e-posta olduğu sonucuna ulaşılır.

Çizelge 2-2 Naive Bayes sınıflandırma ile etiket tahminleme

P(D6|X)=	P(güzel|X)P(kötü|X)P(çok|X)P(kötü|X)=0.5*0.17*0.33*0.17=0.004
P(D6|Y)=	P(güzel|Y)P(kötü|Y)P(çok|Y)P(kötü)=0.1*0.5*0.4*0.5=0.010
P(X|D6)	=P(D6|X)*P(X)/P(D6)=0.0048*0.4/P(D6)
P(Y|D6)	=P(D6|Y)*P(Y)/P(D6)=0.010*0.6/P(D6)

Yukarıdaki örnekte görüldüğü gibi farklı sınıfların çeşitli özelliklere dayalı olasılığını tahmin etmek için benzer yöntem kullanılmaktadır.

Naive Bayes algoritmasının avantajları ve dezavantajları şu şekilde sıralanabilir [11];

- Veri sınıflandırmada hızlı ve kolay uygulanabilir. Çok sınıflı tahminlerde iyi performans sergiler.
- Bağımsızlık varsayımı geçerli olduğunda, Naive Bayes sınıflandırıcısı, lojistik regresyon gibi diğer modellerle karşılaştırmayı daha iyi yapar ve daha az eğitim verisine ihtiyaç duyar.
- Kategorik değişkenlerde, sayısal değişkenlere göre daha iyi sonuç verir.
- Eğitim veri setinde olmayan bir kategorik değişken, test sırasında ortaya çıkar ise Naive Bayes modeli sıfır olasılığı hesaplar ve bir tahmin yapamaz. Bu durum “Sıfır Frenkans” olarak adlandırılmaktadır. Bu durumu çözmek için “Düzeltme Tekniği (Smoothing Technique)” kullanılır. En basit düzeltme tekniği “Laplace Kestirimi”dir. Laplace kestiriminde genellikle olasılığa +1 eklenir.
- Olasılıkların başlangıç değerlerine ihtiyaç duyulur. Bu olasılıkların bilinmemesi durumu genellikle eldeki veriler üzerinden olasılık hesaplanır.
- Gerçek hayatta her zaman bağımsız değişkenler bulmak kolay olmayabilir.

2.1.2 Naive Bayes Kernel

Kernel, parametrik olmayan tahmin tekniklerinde kullanılan bir ağırlıklandırma fonksiyonudur. Kernel, yoğunluk tahmininde rasgele değişkenlerin yoğunluk fonksiyonlarını tahmin etmek için kullanılır [12]. Bunun yanı sıra Kernel Density methodu sınıflandırma için de kullanılabilir [13]. x'in kesintisiz(continuous), y’nin ise k farklı kategoride değer alabilen ayrık bir değer olduğunu varsayalım. n adet $\{x_{i,}, y_i\}$ gözlemi için gelecek gözlemlerde P (y_i = j | x_i) tahmin edilmesi için x'in gözlemlendiği ancak y'nin

gözlemlenmediği bir yöntem elde edilmek istenildiğinde Naive Bayes Kernel ile Eşitlik (4)'deki gibi kolayca hesaplanabilir [13].

$$P(y = j|x_0) = \frac{\pi_j f_j(x_0)}{\sum_{k=1}^{K} \pi_j f_j(x_0)} \quad (4)$$

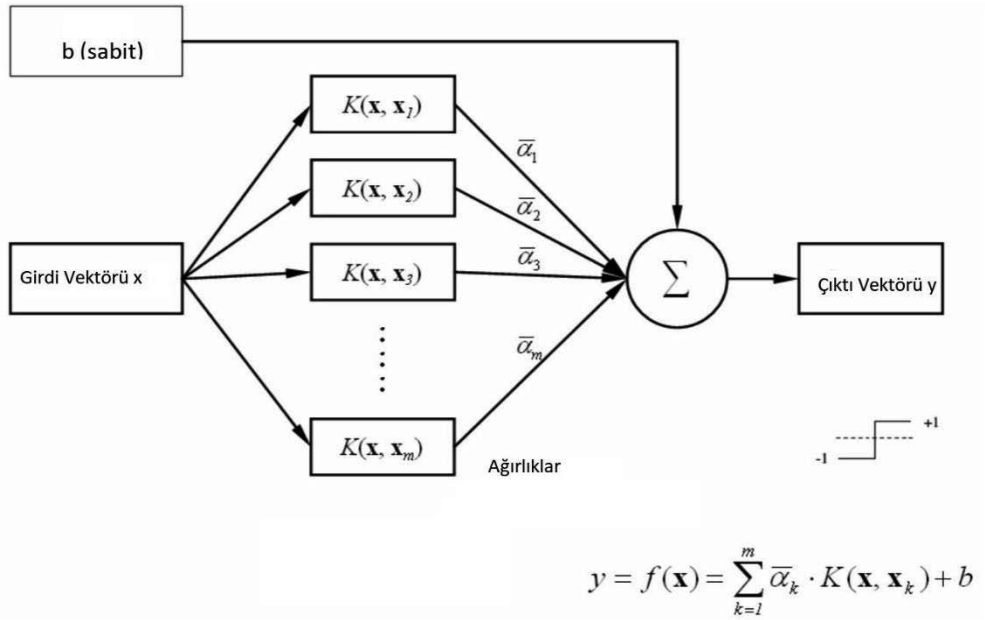
π_j: j sınıfının öncül tahmini(Genellikle π_j, j sınıfı kategorisine giren örnek orandır)

$f_j(x_0)$: yalnızca j sınıfından gelen gözlemleri içeren Kernel yoğunluğuna dayalı x_0'daki tahmini yoğunluğu ifade eder

Naive Bayes sınıflandırıcısının avantajı, sınıflandırma için gerekli değişkenlerin ortalamalarını ve varyanslarını tahmin etmek için yalnızca küçük bir eğitim verisi gerektirmesidir. Bağımsız değişkenler varsayıldığından, tüm kovaryans matrisinin değil sadece her etiket için değişkenlerin varyanslarının belirlenmesi gerekir. Naive Bayes operatörünün aksine, Naive Bayes (Çekirdek) operatörü sayısal niteliklere uygulanabilir [12].

2.2 Destek Vektör Makineleri (SVM)

Vapnik tarafından geliştirilmiş olan SVM, değişkenler arasındaki örüntülerin bilinmediği veri setlerinde sınıflama, regresyon, örüntü tanıma problemleri için kullanılan, istatiksel öğrenme teorisine diğer bir ifadeyle Vapnik-Chervonenkis (VC) teorisine ve yapısal risk minimizasyonuna dayanan, danışmanlı (supervised) öğrenme yöntemidir. SVM, veriye ilişkin herhangi bir birleşik dağılım fonksiyonu bilgisine ihtiyaç duymadığı için dağılımdan bağımsızdır [14]. Şekil 2.3'de SVM'in ağ yapısı verilmiştir [14]. "Burada K(x, x_i) çekirdek fonksiyonlarını, α Lagrange çarpanlarını göstermektedir. Çekirdek fonksiyonları yardımıyla girdilerin iç çarpımları hesaplanmaktadır. Lagrange çarpanları ise ağırlıkları göstermektedir. DVM'de bir örneğe ilişkin çıktı değeri, girdilerin iç çarpımları ile Lagrange çarpanlarının bağımsız kombinasyonlarının toplamına eşittir [14]. "

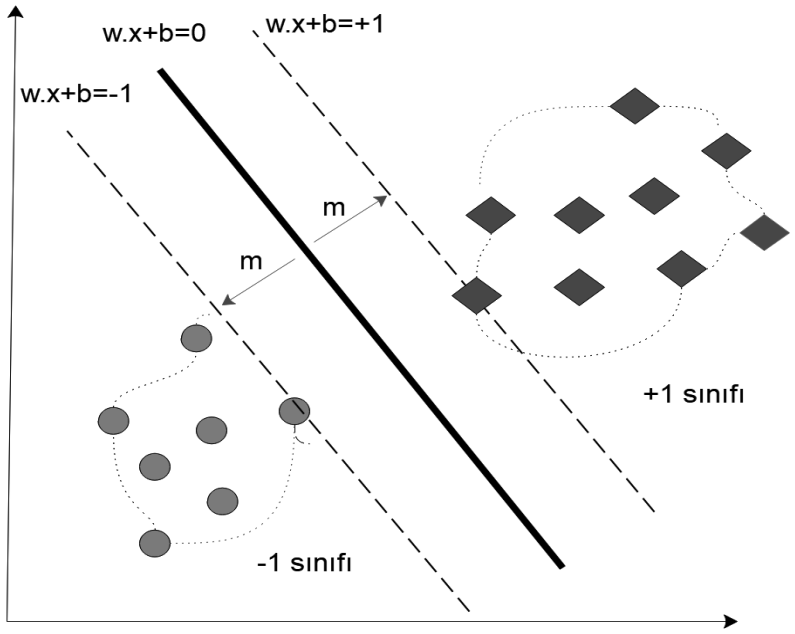


Şekil 2-3 Destek Vektör Makinesi ağ yapısı

SVM'deki temel amaç farklı sınıflara ait vektörleri birbirinden uzaklaştırmak, optimal ayırma hiper düzlemini elde etmektir.

2.2.1 Doğrusal SVM (Linear SVM)

En temel SVM uygulaması olan doğrusal SVM'ler maksimum marjinli çoklu düzlem yaklaşımı ile uygulanırlar. İstenmeyen e-postaların sınıflandırılmasında olduğu gibi 2–sınıflı ve iki boyutlu bir sınıflandırma problemi için doğrusal SVM Destek vektörleri Şekil 2.4'deki gibi gösterilmektedir. Destek vektörleri olarak adlandırılan vektörler ayırma düzlemine en yakın olan her iki sınıfa da ait örnekler olarak gösterilir [14].



Şekil 2-4 2 sınıflı problemler için SVM

Şekil 4.2'de kesikli çizgilerle gösterilen ve ayırıcı çoklu düzleme paralel olarak çizilmiş eşit uzaklıktaki iki çoklu düzlem arasındaki uzaklığa marjin denilmektedir. Eşitlik (5)'de m ile marjin uzaklığı eşitliği gösterilmektedir. Doğrusal SVM kullanılarak x∈p yüksek boyutlu girdi vektörü olmak üzere; (x_i, y_i) ikililerinden oluşan bir eğitim kümesini en iyi ayıracak düzlem Eşitlik (5)'deki gibi hesaplanır:

$$y_i\text{=+1 için } wx_i\text{+b} \geq +1 \qquad (5)$$

$$y_i\text{=-1 için } wx_i\text{+b} \leq -1$$

$$\text{m}=\frac{2}{\sqrt{w*w}} \rightarrow f_{min}(w) = \frac{w*w}{2}$$

x_i: w*x+b=0 çoklu düzlemi üzerinde herhangi bir nokta

y_i : sınıf etiketleri, y_i€{+1,-1}

w: hiper düzlem normali, ağırlık vektörü

m: sınır düzlemi arasındaki uzaklık

b: sabit

2.2.2 Lib SVM

Günümüzde çeşitli sınıflandırma problemlerinde ürettiği iyi sonuçlardan dolayı Destek Vektör Makinesi (SVM), sınıflandırma problemlerinde yaygın olarak kullanılmaktadır. Standart SVM, olasılıksız (non-probabilistic), ikili (binary), doğrusal bir sınıflandırıcıdır. Doğrusal olmayan sınıflandırma yetenekleri eklemek ve sınıfları doğrusal olarak ayrılabilir yapmak için problemi daha yüksek bir boyutsal özellik alanına dönüştüren Kernel fonksiyonlarına ihtiyaç duyulmaktadır. LibSVM, farklı Kernel fonksiyonları, çok katmanlı sınıflandırma ve farklı çapraz doğrulama gibi çeşitli özellikler sunar [15]. Hsu ve Chang tarafından [16] geliştirilmiş olan LibSVM, Destek Vektör Makine'lerinin tüm ayrıntılarını bilmeyen kişiler için SVM'i kolayca uygulamalarına yardımcı olmayı amaçlayan bir hazır kütüphanedir. İlk sürümü çıktığında yalnızca 2 sınıflı problemleri desteklemekteydi. Daha sonra ise geliştirilerek çok sınıflı problemleri ve olasılık tahminleme yetenekleri eklenmiştir [14].

SVM'i bilmeyen kullanıcıların genellikle özellikler ve sınıf etiketlerinden oluşan bir veri kümesini destek vektör makineleri için uygun formata getirirler. Daha sonra ise "linear, polynomial, radial basis function, sigmoid" gibi parametrelerden birini rastgele seçer ve test ederler. Hsu ve Chang ise LibSVM ile bu duruma kolaylık sağlamıştır. LibSVM ile için her bir parametrenin en uygun değeri çapraz doğrulama ile hesaplanmaktadır.

SVM'de sınıflandırma, Kernel matrisinin hesaplanması ve sınıflandırma modelinin oluşturulması olarak iki kısımda ele alınabilir. Giriş verisi büyük boyutlu olduğunda Kernel matrisi hesaplanıp, hafıza da saklanamayan bir hal alır. Bu nedenle çözücü, anında hesaplama yapan işlemci ve bellek bant genişliğine ihtiyaç duyar. Çapraz doğrulama kullanıldığında ise Kernel matrisinin değerlerinin hesaplanması birden fazla kez tekrar edilmelidir. Bu noktada LibSVM kütüphanesinde bulunan "easy.py" komut dosyası varsayılan parametreler ile bu işlemi kolay hale getirir. Sınıflandırma işlemi için hız sağlar [14].

2.2.3 Pegasos SVM

Shai ve Yoram tarafından [17] yapılan çalışma ile ortaya çıkan Pegasos SVM'in açılımı, "Primal Estimated sub-Gradient Solver"'dır. Pegasos, Destek Vektör Makine'lerindeki optimizasyon problemini çözmek için basit ve etkili bir stokastik "sub-gradient descent" algoritması tanımlar. Şekil 2.5'te sözde kodları verilen Pegasos algoritması uygulanırken, her eğitim adımında rastgele bir eğitim verisi seçilir. Seçilen veri için sub-Gradient ile

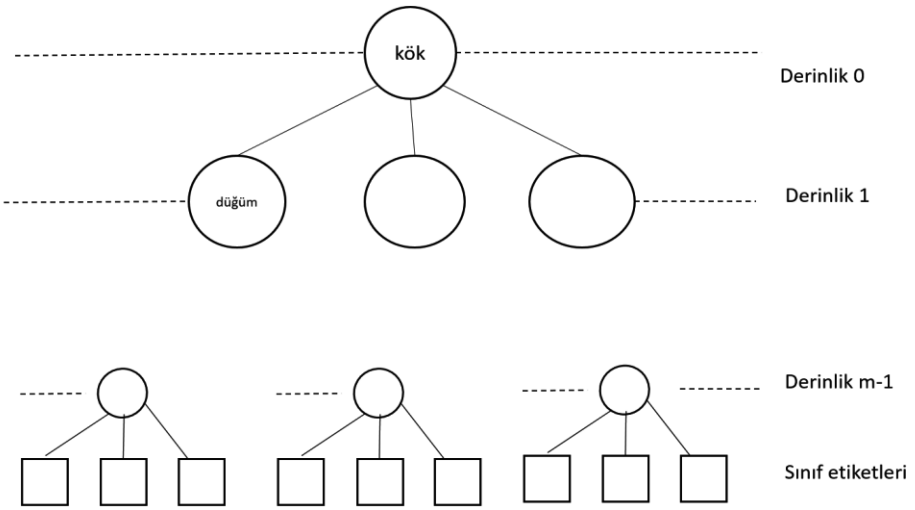
tahminleme yapılır ve önceden belirlenmiş olan adım sayısı kadar ters yönde gidilir. Burada rastgele seçilmiş örnekler üzerinde en yüksek olasılığı göstermek amaçlanmaktadır. Pegasos, geniş metin sınıflandırma problemleri için SVM yöntemlerine kıyasla daha hızlı sonuç vermektedir. Pegasos SVM Algoritması Şekil 2.5'deki gibidir:

INPUT: S, λ, T
INITIALIZE: Set $\mathbf{w}_1 = 0$
FOR $t = 1, 2, \ldots, T$
  Choose $i_t \in \{1, \ldots, |S|\}$ uniformly at random.
  Set $\eta_t = \frac{1}{\lambda t}$
  If $y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1$, then:
    Set $\mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda)\mathbf{w}_t + \eta_t y_{i_t} \mathbf{x}_{i_t}$
  Else (if $y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle \geq 1$):
    Set $\mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda)\mathbf{w}_t$
  [Optional: $\mathbf{w}_{t+1} \leftarrow \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1}\|}\right\} \mathbf{w}_{t+1}$]
OUTPUT: $\mathbf{w}_{T+1}$

Şekil 2-5 Pegasos Algoritması

2.3 Karar Ağaçları

Morgan ve Songuist tarafından oluşturulan algoritmada, sınıfları bilinen örnek veriler basit karar verme adımları ile küçük gruplara bölünür. Her bölme işlemi ile birbirine benzeyen veriler gruplanır ve tümevarım yöntemi ile sınıflandırma yapılır [18]. Karar ağaçlarının genel görünümü Şekil 2.6 [18]'daki gibidir.



Şekil 2-6 Karar ağacı örneği

Karar ağaçlarında temel adım karar düğümlerini oluşturmaktır. Karar düğümlerinin oluşturulurken ağacın dengeli bir şekilde dallanması ve sınıflandırma işleminin doğru yapılabilmesi için en iyi nitelik düğüm olarak seçilmelidir. Bunun için Shannon ve Weaver tarafından ortaya çıkarılan "Bilgi Kazancı Teorisi" ile bütün sistemdeki beklenen değer hesaplanır [19]. Bilgi kazancı Eşitlik 6'daki gibi hesaplanır.

$$H=-\sum_{i=1}(p_i \log p_i) \quad (6)$$

$$\text{Gain(S, A)=H(S) - } \sum_{v\in Values(A)} \frac{|S_v|}{S} H(S_v)$$

p_i:i sınıfının öncül olaşılığı

H: Entropy

S: Örnek uzay

S_v : Örnek uzay alt kümesi

Danışmanlı öğrenme algoritmaları içerisinde oldukça yaygın olan karar ağaçlarının algoritma mantığı Çizelge 2.3'deki gibidir:

Çizelge 2-3 Karar Ağaçları Algoritması

Veri Seti OluşturmaAlgoritması

Girdi: T (Öğrenme Kümesi)
1. Her nitelik için bilgi kazancı hesapla
2. Bilgi kazancı en yüksek niteliği "ayırıcı nitelik" seç
3. T'yi ayırıcı nitelik ile böl, düğüm oluştur.
If(örnekler aynı sınıfa ait || örnekleri bölecek nitelik yok|| kalan nitelik için örnek yok)
{
İşlemi sonlandır
}
else
{
1. adıma geri dön
}

Karar ağaçlarının sürekli ve ayrık değerler için kullanılabilir olması, ağaç oluşturmadaki kolaylık, kuralların kolay anlaşılabilir olması gibi avantajları vardır. Diğer yandan sürekli olarak nitelik tahmin etmek karar ağaçları için zordur. Çok sınıflı verilerde avantajlı değildir [17].

"Karar ağacı oluşturmak için geliştirilen algoritmalar arasında CHAID (Chi- Squared Automatic Interaction Detector), Exhaustive CHAID, CRT (Classification and Regression Trees), ID3, C4.5, MARS (Multivariate Adaptive Regression Splines), QUEST (Quick,

Unbiased, Efficient Statistical Tree), C5.0, SLIQ (Supervised Learning in Quest), SPRINT (Scalable Paralleizable Induction of Decision Trees) gibi algoritmalar yer almaktadır [17].”

2.4 Yapay Sinir Ağları

İlk olarak1943’lı yılların başında Warren McCulloch ve W.A Pitts tarafından atılan, İnsan beyninden esinlenerek geliştirilmiş olan Yapay Sinir Ağları(YSA) eldeki verilerden yola çıkarak genelleme yapmakta ve hiç görmediği veriler hakkında bu genellemelere göre karar verebilmektedir. YSA’lardaki her bir sinir kendi bilgi işleme yapısına sahiptir ve diğer sinirlerle ağırlıklı bağlantılar aracılığıyla birbirine bağlanır [20].

YSA’ların diğer öğrenme algoritmalarından farkları şu şekilde sıralanabilir:

- Diğer algoritmalar “belirlenen kurallara girişler uygulandığında çıkışlar oluşur” mantığını temel alırken YSA’lar giriş-çıkış bilgileri verilerek kuralları koyar.
- YSA’lar deneyimlerden yararlanır. Diğer algoritmalarda bilgiler kesindir.
- Hesaplama; toplu, eş zamansız ve öğrenmeden sonra paraleldir.
- Diğer algoritmalara göre daha yavaştır.
- Diğer algoritmalarda bellek paketlenmiş ve hazır bilgi depolanmış ilen YSA’larda bellek ayrılmış ve ağa yayılmıştır.
- Hata toleransı diğer algoritmalarda yok isek YSA’da vardır.

Şekil 2.7 [20]’de yapay bir sinirin bölümleri gösterilmiştir.

Girişler: (X_1 , X_2 ,….. X_n) Çevreden aldığı bilgiyi sinire getirir. Girişler kendinden önceki sinirlerden ya da dış dünyadan sinir ağına eklenebilir.

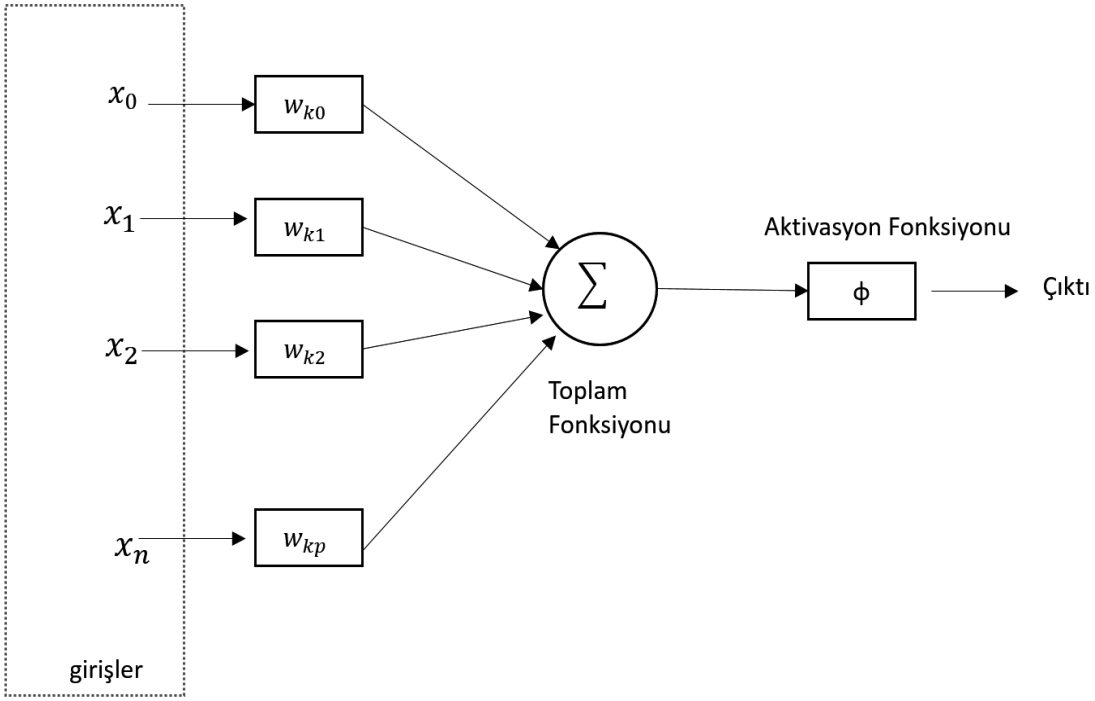
Ağırlıklar: (w_1 , w_2 ,….. w_n) yapay sinir tarafından alınan girişlerin sinir üzerindeki etkisini belirleyen uygun katsayılardır. Her bir girişin kendine ait bir ağırlığı vardır.

Toplama fonksiyonu: Sinirde her bir ağırlığın ait olduğu girişlerle çarpımının toplamlarını bir eşik değeri ile toplayarak aktivasyon fonksiyonu (etkinlik işlevi)ne gönderir. Bazı durumlarda toplama fonksiyonu en az (min), en çok (max), çoğunluk veya birkaç normalleştirme algoritması gibi çok daha karmaşık olabilir.

Aktivasyon fonksiyonu(etkinlik işlevi): Toplama işleminin sonucu etkinlik işlevinden geçirilip çıkışa iletilir. Etkinlik işlevinin çıkışı y_i, giriş vektörleri x_i tarafından uyarlandığında Eşitlik (7)’deki gibi tanımlanır.

$$y_i = \begin{cases} 1 \text{ eğer } w_1x_1 + w_2x_2 \ldots\ldots + w_nx_n \geq T \\ 0 \text{ eğer } w_1x_1 + w_2x_2 \ldots\ldots + w_nx_n \leq T \end{cases} \quad (7)$$

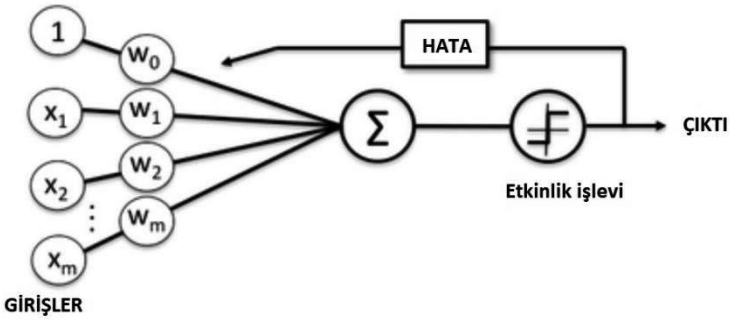
İkili girişlerin bir örneği verildiğinde, etkinlik işlevi sıfır ya da bir çıkışını verecektir.



Şekil 2-7 Yapay sinir ağı

2.4.1 Perceptron

"F. Rosentblatt tarafından 1960'lı yıllarda sinir ağları olarak önerilmiştir. Rosentblatt, Perceptronları sıradan makinelerin çözemediği problemleri çözebilen "yeni bir tür bilgisayar" olarak ifade etmekteydi [20]" Perceptron, beyin işlevlerini modelleyebilmek amacıyla yapılan çalışmalar neticesinde ortaya çıkan tek katmanlı eğitilebilen ve tek çıkışa sahip olan yapay sinir ağıdır [21]. Birden fazla girişi alıp tek bir çıkış üretilmesinden sorumludur. Şekil 2.8'deki gibi gösterilebilir. "Perceptron doğrusal bir fonksiyonla iki parçaya bölünebilen problemlerde kullanılabilir. Bu problemlere AND, OR, NOT durumları örnek olarak verilebilir [22]."



Şekil 2-8 Perceptron örneği

Perceptron, girişlerin ağırlıklı toplamını Eşitlik (7)'de belirtilen eşik değeri T ile karşılaştırır. Ağırlıklı toplam eşik değerinden büyük ise sonuç 1 aksi durumda 0'dır.

2.5 En Yakın Komşu K-NN

K En Yakın Komşu (K-NN) Algoritması, T. M. Cover ve P. E. Hart tarafından önerilen, temelde sınıflandırılmak istenen verinin, daha önceki verilerden k tanesine olan yakınlığına bakma mantığına dayanan bir sınıflama algoritmasıdır [23]. K rasgele bir değerdir. Sınıflandırılmak istenen veri, daha önceden sınıflandırılmış olan ya da diğer bir değişle daha önceden etiketi belli olan verilerden kendisine en yakın k tanesini el alır. Bu elemanlar hangi sınıfa ait ise sınıflandırılacak eleman o sınıfa ait olur. Elemanlar arasındaki mesafe genelde "Öklid mesafesi" ya da "Manhattan mesafesi" ile hesaplanır.

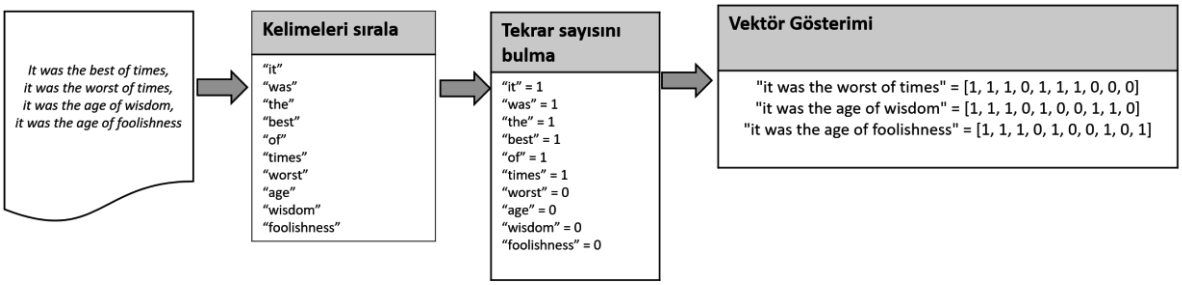
En Yakın Komşu Algoritmasında sınıflandırma performansını doğrudan etkileyen parametrelerden birisi "K" değeridir [24]. Bulut ve Amasyalı tarafından yapılan çalışmada k değeri değiştirilerek k değerinin sınıflandırma performasa olan etkileri gözlemlenmiştir. K değeri hassas bir değerdir. Bu nedenle bazı sınıflandırma çalışmalarında K'nın artırılması performansa olumlu etkide bulunmuş iken bazılarında tam tersi etki göstermiştir. Bu nedenle hassas bir şekilde k parametresi seçilmelidir.

K-NN Algoritması analitik olarak izlenebilir olması ve kolay olması, aykırı verilerden etkilenmemesi gibi nedenlerle birçok yerde tercih edilmektedir.

2.6 Kelime Kümesi Tekniği (Bag of Words)

Metin verilerinden öğrenme yaparken en önemli problem her bir dokümanın farklı uzunlukta olmasıdır. Böyle durumlarda her bir kelimeyi bir özellik olarak ele almak gerekir. Oysaki bu durumu uygulamak büyük çaplı verilerde oldukça zordur. Bu durumun üstesinden gelebilmek için en önemli yöntemlerden birisi ise Kelime Kümesi Tekniği (Bag of Words) (BOW)'dir. Bu yöntemde her bir kelime benzersiz olarak ele alınır ve her bir benzersiz kelimenin,

doküman içerisinde tekrar etme sayısı bulunur. Böylece makine öğrenme algoritmaları modellemede kullanılmak üzere metinden özellik çıkarımı yapılır [24][25]. Kullanımı aşağıdaki örnekteki gibidir [26]:



Şekil 2-9 BOW yöntemi örneği

Bu örnekte ilk olarak dokümandaki benzersiz kelimelerin listesi çıkarılır. Daha sonra bu kelimelerin doküman içerisindeki tekrar sayıları hesaplanır. Toplamda 10 adet benzersiz kelime vardır. Bu tüm vektörlerin boyutunun 10 olacağı anlamına gelmektedir. Her bir cümle için vektör gösterimi Şekil 2,9'daki gibidir.

N gram ise verilen bir dizi kelime içerisinde tekrarları bulmaya ve olasılıkları kelimelere atamaya yarar. N gramlı bir karakter n ardışık karakter ya da kelime dizisidir [27]. Eşitlik 8'deki gibi hesaplanır.

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_{k-N-1}^{k-1}) \quad (8)$$

N bir olduğunda unigram, iki olduğunda bigram, üç olduğunda ise trigram olarak adlandırılır. Bir cümle için trigram oluşturma aşağıdaki örneklenmiştir:

Örnek: Esra Şahin Hacettepe Üniversitesi Bilgisayar Mühendisliği

Trigram'lar: ["Esra Şahin Hacettepe", "Şahin Hacettepe Üniversitesi", "Hacettepe Üniversitesi Bilgisayar", "Üniversitesi Bilgisayar Mühendisliği"

3. ALANYAZIN ÖZETİ

İstenmeyen e-posta / e-posta sınıflandırma teknikleri üzerine birçok çalışma sunulmuştur. Araştırmalarının çoğunda araştırmacılar, istenmeyen e-postaları ayırt etmek için makine öğrenme algoritmalarının eğitiminde kullanılacak olan içerik temelli yaklaşımlara yönelmişlerdir. Bu içerik temelli yaklaşımlar 4 ana kategoriye ayrılabilir. Bunlar; başlık (header) özellikleri, konu (subject) özellikleri, e-posta metni(body) özellikleri ve e-posta eki (attachment) özellikleridir [28].

Yapılan çalışma örneklerini sırasıyla inceleyecek olursak Sah ve arkadaşları [28] tarafından yapılan çalışmada sırasıyla veri toplama işlemi, verileri ön işleme, verilerden özellik çıkarma, verileri sınıflandırma ve sonuçları analiz etme işlemi yapılarak metin(body) tabanlı sınıflandırma çalışması yapılmıştır. Çalışmada eğitim için 702 e-posta, test için 260 e-posta kullanılarak Support Vector Machine (SVM) ve Naive Bayes algoritmaları ile sınıflandırma yapılmıştır. Çalışmada yapılan özellik çıkarımı aşaması bu tez çalışmasında özellik çıkarımı için yapılan işlemlere benzerlik göstermektedir. Çalışma sonucunda Naive Bayes ve SVM algoritmalarının benzer başarı gösterdiği sonucuna varılmıştır. Benzer bir diğer çalışma Renuka ve arkadaşları [29] tarafından Hadoop ortamında yapılan sınıflandırma çalışmasında istenmeyen e-postaları ayırt etmek için Gradient Boost ve Naive Bayes sınıflandırma tekniği kullanılmıştır. Çalışma temelde eğitim ve test olmak üzere iki aşamada ele alınmıştır. Çalışma da sınıflandırma performansını iyileştirmek için tek düğüm Hadoop ortamı kullanılarak farklı sınıflandırma tekniklerinin bir arada kullanıldığı karma bir model çıkarılmıştır. Duyarlılık (precision), doğruluk (accuracy) ve hassasiyet (recall) metrikleri ile çalışmanın performansı ölçülmüştür.

İstenmeyen e-posta/e-posta ayrımında özellik çıkarımı sınıflandırma başarısını doğrudan etkileyen işlemlerdir. İstenmeyen e-postalarda genellikle sınıflandırmaya çalıştığımız nesneler metinlerdir. İstenmeyen e-postaların metin kısımlarından özellik çıkarımı istenmeyen e-posta göndericilerin tekniklerini sürekli değiştirmesinden ötürü istenmeyen e-postalardaki özellikler de sık sık değişmektedir.

Diğer yandan özellik çıkarımı aşamasında bir nevi etiketli veriler oluşturulmaktadır. Etiketli verileri kullanarak eğitim yapmak, performansın korunmasına yardımcı olsa da, etiketli verileri sürekli hazırlamak oldukça pahalıdır. Kumagai ve arkadaşları [30] tarafından yapılan çalışmada öğrenme işlemi için etiketli verilerin yanı sıra daha az maliyet getirecek şekilde etiketsiz verilerde kullanılarak sınıflandırma yapılmıştır. Çalışmadaki temel kazanım eğitim aşamasında hiç görülmemiş bir özelliğin daha sonra ortaya çıkarak performansa olan olumsuz etkilerini ortadan kaldırmaktır. Diğer bir önemli kazanım ise eğitim ve test evreleri arasındaki dağılımın değişmesinden doğan performans kötüleşmesini iyileştirmektir. Önerilen yöntem her iki sorunun üstesinden gelen en son sınıflandırıcıları öğrenir. Önerilen yöntemle, mevcut özellikler göz önüne alındığında yeni özelliklerin koşullu dağılımı etiketsiz veriler kullanılarak öğrenilmiştir. Deneylerde önerilen yöntemin var olan mevcut yöntemlerden daha iyi performans gösterdiğini, yeni özelliklerin bulunduğu, eğitim ve test dağılımlarının farklı olduğunu anlatılmıştır.

Özellik çıkarımına makine öğrenme teknikleri açısından bakıldığında aritmetik nesneler (rakamlar ya da vektörler) metinlere göre daha iyi kategorize edilebilir. Ajaz ve arkadaşları tarafından yapılan istenmeyen e-posta/e-posta sınıflandırma çalışmasında [31] öncelikle tüm mesajlar sayılardan oluşan özellik vektörlerine dönüştürülmüştür. Çalışmada özellik vektörleri çıkarılırken çoğu kez bilgi kaybedildiği, özellik çıkarıcıyı tanımlama şeklinin sınıflandırma başarısı için önemli olduğu vurgulanmıştır. Ayrıca özellik seçimini yapılırken özelliklerin hepsinin mesaj metninden alınmaması gerektiği, aynı zamanda özellik çıkarma sürecine kendi çıkardığımız bilgilerin de eklene bilineceği söylenmiştir. Çalışmada SHA 512 algoritması ve Naive Bayes sınıflandırma metodu kullanılmıştır.

E-postaların özellikle metin (body) kısımlarına yoğunlaşan sınıflandırma çalışmalarının yanı sıra konu (subject) bilgisi ile sınıflandırma yapan çalışmalarda mevcuttur. Lee ve arkadaşları tarafından yapılan çalışmada [32] ağırlıklı Naive Bayes sınıflandırma metodu temel alınarak bir istenmeyen e-posta filtresi geliştirilmiştir. Geliştirilen filtre sadece istenmeyen e-postaların konusunu kontrol etmektedir. Çalışma yapılırken sık kullanılan kelimeleri çıkartmak için Bag of Words (BOW) tekniği yanı sıra e-posta konularındaki yeni karakteristikleri keşfetmek için doğal dil işleme teknikleri de kullanılmıştır. Çalışmada diğer çalışmalarda sıkça kullanılan istenmeyen e-posta/e-posta veri setleri(TREC, EnronSpam [32]) kullanılmıştır. Bahsedilen yöntem %94.85 doğruluk oranı ile sınıflandırma yapılmıştır. Çalışma aynı türdeki diğer çalışmalara göre %2.43 daha iyi başarı sağlamıştır.

İstenmeyen e-posta sınıflandırma çalışmalarında SVM ve Naive Bayes algoritmalarının popüler olduğunu görmekteyiz. Carreras ve Marquez tarafından [33] yapılan çalışmada ise bunlardan farklı olarak AdaBoost algoritması kullanılarak istenmeyen e-posta filtrelemesi yapılmıştır. Çalışmada açık veri setlerinden PU1 [33] kullanılmıştır. Çalışma sonucunda Boosting temelli methodların PU1 veri setleri için Naive Bayes ve Karar Ağaçları'nın çıkartıldığı yöntemlerden daha üstün başarı gösterdiği görülmüştür. Diğer yandan yüksek hassasiyetli sınıflandırıcıların gerekli olduğu senaryolarda AdaBoost sınıflandırıcının düzgün çalıştığı kanıtlanmıştır. Karmaşıklığın artırılması daha yüksek hassasiyetli sınıflandırıcılar elde etmeyi mümkün kılmaktadır. Karmaşıklık maliyeti artırmış olsa dahi bu maliyet yanlış sınıflandırma maliyetinden daha düşük olacaktır. Sınıflandırma çalışmalarında tercih edilen bir diğer algoritma ise Yapay Sinir Ağları (ANN)dır. Chuan ve arkadaşları [34] tarafından yapılan çalışmada açık kaynaklı e-posta veri seti kullanılarak ANN-LVQ ve ANN-BP yaklaşımı ile istenmeyen e-postalar içeriklerine göre sınıflandırılmışlardır. Çalışmada sinir ağı eğitimi sayısının sınıflandırma başarısını doğrudan

etkilediği, bu sayı 1500'e ulaştığında performansın iyi duruma geldiği vurgulanmıştır. Çalışma da yapay sinir ağı temelli algoritmaların istenmeyen e-posta/e-postalara dair her bir özelliğin bütünle ilişkisini gözetmiş olmasından dolayı, her bir özelliği birbirinden bağımsız olarak ele alan Bayes algoritmalarına göre daha iyi olduğu savunulmuştur. Deney sonuçlarına bakıldığında ANN BP algoritmasının %98,42, ANN-LVQ algoritmasının %98.97 başarıya ulaştığı görülürken Naive Bayes algoritmasının %97.63 başarıya ulaştığı görülmüştür. Sharma ve Sahni [35] tarafından yapılan çalışmada ise ID3, J48, Simple CART, Alternating Decision Tree (ADT) algoritmaları kullanılarak Weka ortamında istenmeyen e-posta verileri sınıflandırılmıştır. Çalışmada UCI Makine Öğrenmesi deposundan alınan veri setleri kullanılmıştır. Çalışma sonucunda ID3 algoritması için %89, J48 algoritması için %92, ADT algoritması için %90,91, SimpleCART algoritması için %92.63 başarıya ulaşmıştır. J48 sınıflandırıcı sınıflandırma doğruluğu açısından ID3, CART ve ADTree'den daha iyi performans gösterdiği görülmüştür.

Alanyazındaki çalışmalar ve bu çalışmalarda yer alan bilgilere göre istenmeyen e-posta/ e-posta sınıflandırma çalışmaları ile ilgili genel olarak şu sonuçlar çıkartılmaktadır;

- Kötü amaçlı istenmeyen e-posta araştırmalarının çoğunda araştırmacılar, istenmeyen e-postaları ayırt etmek için makine öğrenme algoritmalarının eğitiminde kullanılacak olan içerik temelli yaklaşımlara yönelmişlerdir. Bu yaklaşımlar 4 ana kategoriye ayrılabilir. Bunlar; başlık (header) özellikleri, konu (subject) özellikleri, e-posta metni(body) özellikleri ve e-posta eki (attachment) özellikleridir [28].
- Çalışmalarda en çok kullanılan makine öğrenmesi teknikleri; SVM, ANN, RF, Naive Bayes ve AdaBoost olmuştur [28].
- Bazı araştırmalarda ise bilinen makine öğrenme tekniklerinin yanı sıra Rough setler gibi kural bazlı yaklaşımları da konu alan karma yaklaşımlar sergilenmiştir [28].
- Çalışmalarda genel olarak kullanılan e-posta verileri Spambase(1999), Spam Assassin(2006), TREC(2007)'dir [28].

4. MATERYAL VE YÖNTEM

Bu bölümde İstenmeyen e-posta/e-posta sınıflandırma yöntemi detaylı olarak ele alınmaktadır. Bölüm içerisinde sırasıyla veri işleme, araç seçimi, deneylerin yapılması işlemleri anlatılacaktır.

4.1 Veri Ön İşleme

Tez çalışması kapsamında kullanılan tüm veriler, istenmeyen e-postalar / e-postalarda yer alan bağlantı(link) bilgilerini ve bu bağlantılarda yer alan metin bilgilerini, benzersiz bir e-posta numarasını içermektedir. İşlenmemiş veriler Çizelge 4.1'deki formatta sunulmuştur.

Çizelge 4-1 İşlenmemiş veri formatı

	FORMAT
E-posta	**Benzersiz E-Posta Numarası** ||| E-Posta Bağlantısı(URL) ||| *Bağlantı Metni* Örnek: **00000460-6b36-41c3-aeaf** **35bb32c02766**|||http://www.bigbv.top/5D899TU358EM391XL1721W2346IN18T678649B3249141020.php|||*Click Here*
İstenmeyen E-posta	**Benzersiz E-Posta Numarası** ||| İstenmeyen E-Posta Bağlantısı(URL) ||| *Bağlantı Metni* Örnek: **0002d83c-90c4-48ab-80ea-** **594c411463db**|||http://www.totalspas.top/911/362/395/1734/2360.19tt1319463AAF3.php|||*Start Building Amazing Sheds The Easier Way With A Collection Of 12,000 Shed Plans!*

İşlenmemiş veri seti aşağıdaki özelliktedir;

- E-posta numarası, bağlantı ve bağlantı metinleri "|||" karakterleri ile birbirinden ayrılmaktadır.
- E-posta numaraları harf ve rakamlardan oluşan benzersiz numaralardır.
- Bir e-posta içerisinde birden fazla bağlantı bulunabilmektedir. Bu durum benzersiz e-posta numaraları ile ayırt edilebilmektedir.
- Bağlantılar özellik çıkarımı için uygun değildir. Her bağlantı farklı uzunlukta ve farklı formattadır.
- Bağlantı metinleri dilden bağımsızdır. Bunun yanı sıra İngilizce ve Türkçe metinlerin çoğunlukta olduğu görülmüştür.

- Bağlantı metinlerinde noktalama işaretlerinin sıkça yer aldığı görülmüştür. Bu durum makine öğrenme tekniklerinin performans başarımını artırmak için istenmeyen bir durumdur.

İşlenmemiş verilerin özelliklerine göre aşağıdaki gereksinimler çıkarılmıştır.

- Her bir satır verinin benzersiz e-posta numarası, bağlantı ve bağlantı metni olarak parçalanması gerekmektedir.
- Bağlantı metinlerinde yer alan rakamların ve tanımlanamayan karakterlerin silinmesi gerekmektedir. Silme işlemi sırasında karakterlerin silindiği noktaya boşluk eklenecektir. Bu boşluklar kelimelerin ayırılması için yardımcı olacaktır. Ancak kesme işareti için özel durum vardır. Kesme işareti kaldırılırken kelimenin ekinin ve kökünün ayrı iki kelime gibi değerlendirilmemesi için kesme işareti yerine boşluk konulmayacaktır.
- İki karakter olan kelimeler bir anlam taşımadığı için silinmesi gerekmektedir.
- Bazı özel anlam içeren kelimelerin ayırt edici niteliği olmadığı için çıkarılması gerekmektedir. Bunlar; Türkçe ve İngilizce haftanın günleri, günlerin kısaltması, Türkçe ve İngilizce yılın ayları, ayların kısaltması, para birimi kısaltmaları, "WWW, http, HTTPS, COM, AND, WITH, THE, İLE, VIA, ANY, YOU, THEY, ARE vs." gibi kelimelerdir. Bu kelimelere veri seti incelenerek karar verilmiştir.
- Türkçe ve İngilizce alfabelerinin farklılığından dolayı oluşan hataları en aza indirmek için tüm harfler büyük harfe çevrilecektir. Türkçe de yer alan noktalı harfler, noktasız hale dönüştürülecektir. Bu durum aynı kelime olup birinde noktalı, diğerinde noktasız yazılmış karakterler nedeniyle kelimelerin farklı algılanmasının önüne geçecektir.

Veri ön işleme aşamasının gerçekleştirilebilmesi için Windows 10 işletim sistemi ortamında, 16 Gb RAM, 2.60 GHz işlemci hızına sahip bilgisayar kullanılmıştır. Yukarıda sıralanan gereksinimlerin gerçekleştirilebilmesi için Visual Studio 2015 ortamında C# programlama dili kullanılarak uygulama geliştirilmiştir. Uygulamanın sözde kodları Çizelge 4.2'deki gibidir;

Çizelge 4-2 Veri ön işleme algoritması sözde kodları

Veri Önişleme Algoritması

```
Girdi: D:İstenmeyen e-posta/eposta veri seti (“e-posta numarası|||bağlantı|||bağlantı metni” formatında)
Çıktı O: İşlenmiş veri seti
foreach (verisatırı in D)
{
    “|||” işaretine göre ayır;
    verisatırı=verisatırı- e-posta numarası;
    bağlantı metni=verisatırı- bağlantı;
     If(bağlantı metni!=null)
    {
      Bağlantı metnini büyük harfe çevir;
      Türkçe karakterleri noktasız karakterlere çevir;
      Özel karakterleri kaldır, yerine boşluk koy;
        If(Özel karekter “ ‘ “ ise)
        {
           Kesme işaretini kaldır, boşluk koyma;
        }
       Kelime listesi=bağlantı metnini kelimelere ayır;
     }
   Foreach( kelime in kelime listesi)
  {
     If(kelime uzunluğu>2)
      {
         If(kelime!= özel kelime)
          add tek kelime listesi
       }

   }
    Foreach( tek kelime in tek kelime listesi)
    {
      Temizlenmiş bağlantı metni+=tek kelime
    }
Temizlenmiş bağlantı metni listesi=temizlenmiş bağlantı metni
}
O=temizlenmiş bağlantı metni listesi
```

İstenmeyen e-postalar ve e-postalar için ayrı dosyalar üzerinde veri ön işleme işlemi gerçekleştirilmiştir.

Bir e-postada birden fazla bağlantı bulunabilir. Aynı e-posta içerisinde yer alan bağlantılar ve bağlantı metinleri birbirinin aynısı olabilir. Farklı örneklerin görülebilmesi amacıyla bir epostadaki bağlantı ve bağlantı metni aynı olan e-postadan sadece bir tanesi işleme alınmıştır.

Bu ayırt etme işleminin yapılabilmesi için Çizelge 4.3’de sözde kodları verilen algoritma kodlanmıştır.

Çizelge 4-3 Mükerrer Verileri Ayırt Etme Algoritması sözde kodları

Mükerrer Verileri Ayırt Etme Algoritması

```
Girdi: D:İşlenmiş e-posta/eposta veri seti (“e-posta numarası|||bağlantı|||bağlantı metni” formatında)
Çıktı O: Mükerrer kayıtlardan temizlenmiş veri seti
M: Mükerrer bulunmayan liste=null; //e-posta numarası ve bağlantı metnini tutar.
Temp=true;
Nesne=null; //e-posta numarası ve bağlantı metni özelliklerine sahiptir.

foreach (verisatırı in D)
{
    “|||” işaretine göre ayır;
    e-posta numarası=verisatırı ilk “|||”e göre ayrılmış veri
    verisatırı=verisatırı- e-posta numarası;
    bağlantı metni=verisatırı- bağlantı;

    N.epostanumarası=e-posta numarası;
    N.bağlantımetni=bağlantı metni;
    If(temp)
   {
    M[0]+=N;
    N=null;
    temp=false;
    continue;
    }

     If(bağlantı metni!=null)
    {
      If(M contains N==false)
        M.add(N);
      N=null;
     }
}
O=M;
```

Bu işlemlerin sonucunda istenmeyen e-posta ve e-postalara ait linklerdeki bağlantı metinleri ayırt edilmiş ve ön işleme tabi tutulmuştur. Ön işlem aşaması sayesinde bağlantı metinlerinin hepsi büyük harfe çevrilmiş, yabancı karakter sorunu çözülmüş, noktalama işaretleri kaldırılmış, bazı özel anlam içeren kelimelerin ve 3 harften daha kısa olan kelimelerin ayırt edilmesi sağlanmıştır. Ön işlem sonucunda “.txt” uzantılı istenmeyen e-posta ve e-postalara ait iki adet veri dosyası oluşturulmuştur.

Ön işlem öncesi ve ön işlem sonrası veri sayıları Çizelge 4.4’deki gibidir;

Çizelge 4-4 Veri ön işleme aşaması sonrası veri sayısı durumu

	Ön İşlem Öncesi Satır Sayısı	**Ön İşlem Sonrası Veri Satır Sayısı**	**Ön İşlem ile Silinen Veri Satır Sayısı(fark)**
E-posta Veri Satırı Sayısı	141414	107163	34251
İstenmeyen E-posta Veri Satırı Sayısı	150000	132254	17746

Bir e-postada birden fazla bağlantı bulunabilmektedir. Ön işlem öncesi ve ön işlem sonrası e-posta sayıları Çizelge 4.5'deki gibidir;

Çizelge 4-5 Veri ön işleme aşaması sonrası e-posta sayısı durumu

	Ön İşlem Öncesi E-posta Sayısı	**Ön İşlem Sonrası E-posta Sayısı**
E-posta	8506	8500
İstenmeyen E-Posta	47382	47213
Toplam	55888	55713

4.2 N Gram Özelliklerin Çıkartılması

Ön işlemden geçirilmiş olan verilerin, makine öğrenmesi yöntemlerinde kullanılması için eğitim ve test işlemlerinde kullanılacak veri seti haline getirilmesi gerekmektedir. Bu amaçla Kelime Kümesi Tekniği kullanılarak verilerin 1 gram, 2 gram, 3 gram, 4 gram ve 5 gram olmak üzere özellikleri (*feature*) çıkarılmıştır.

Özellik çıkarma işlemi için Windows 10 işletim sistemi yüklü olan, 16 Gb RAM, 2.60 GHz işlemci hızına sahip bilgisayar ortamında Visual Studio 2015 geliştirme aracı ve C# programlama dili kullanılarak uygulama geliştirilmiştir. N Gram Algoritması'nın sözde kodları Çizelge 4.6'daki gibidir;

Çizelge 4-6 N Gram Algoritması sözde kodları

N Gram Algoritması

```
Girdi: D: Ön işlemden geçmiş veri seti (bir yada daha fazla kelimeden oluşmuş cümle formatında)
         N: N gram boyutu
Çıktı O: N gram
foreach (verisatırı in D)
{
      Tek kelime listesi=veri satırını kelimelere ayır;
}
    for (i=0; i<=tekkelimelistesi eleman sayısı-N;i++)
 {    switch(N)
      {
      case 1:
       NGram+=tekkelimelistesi[i];
       break;
      case 2:
       NGram+=tekkelimelistesi[i]+" "+ tekkelimelistesi[i+1];
       break;
       case 3:
       NGram+=tekkelimelistesi[i]+" "+ tekkelimelistesi[i+1] +" "+ tekkelimelistesi[i+2];
       break;
       case 4:
       NGram+=tekkelimelistesi[i] +" "+ tekkelimelistesi[i+1] +" "+tekkelimelistesi[i+2]+" "+
tekkelimelistesi[i+3];
       case 5:
     NGram+=tekkelimelistesi[i] +" "+ tekkelimelistesi[i+1] +" "+ tekkelimelistesi[i+2]+" "+
tekkelimelistesi[i+3] +" "+ tekkelimelistesi[i+4];
        break;
      }
 add NGram Listesi (Ngram)
  }
O=NGram Listesi
```

İstenmeyen e-postalar ve e-postalar için ayrı ayrı özellik çıkarım işlemi yapılmıştır. Ayrı ayrı oluşturulan bu özellik kümeleri birleştirilerek ortak olan gramlardan tek bir tanesi özellik olarak sayılmıştır. Oluşturulan her özellik, diğer bir deyişle her bir gram, makine öğrenme tekniklerinde kullanılacak olan matris yapıdaki veri setlerinin sütunlarını oluşturmaktadır. Gramlar oluşturulduktan sonra ortaya çıkan sonuçlara bakıldığında özellik sayılarının oldukça fazla olduğu görülmüştür. Her bir gramın işlenmiş veri setinde kaç kez geçtiği hesaplanmıştır. Çıkan sonuçlara göre belirli limitlere göre özellik sayısının azaltılması işlemi yapılmıştır. Özellik sayılarının azaltılması 30 tekrar, 40 tekrar ve 50 tekrar sayısı limit olarak belirlenmiş, bu sayılardan daha az tekrar eden gramlar özellik kümesine dâhil edilmemiştir. Özellik kümesinin bu şekilde tasarlanması aynı zamanda özellik sayısının makine öğrenme teknikleri başarısına olan etkisini gösterecektir. Limit eklemeden ve 30, 40, 50 tekrar limitlerine göre oluşturulan özelliklerin sayıları Çizelge 4.7'deki gibidir. Burada istenmeyen e-postalara ait özellik sayısı ve e-postalara ait özellik sayıları verilmiştir. Toplam olarak

verilen özellik sayısı ise veri setlerinde kullanılacak olan özellik sayısını göstermektedir. Toplam sayısı çıkartılırken istenmeyen e-posta özellikleri, e-posta özellikleri birleştirilmiş, her iki kümede olan ortak özellikler bir kez sayılmıştır.

Çizelge 4-7 Limitlere göre özellik sayıları

		1 Gram	2 Gram	3 Gram	4 Gram	5Gram
LİMİT YOK	İst. EP	19450	82115	127040	171669	214554
	EP	35741	114861	147189	168251	183528
	Toplam	48580	193049	272615	339069	397587
LİMİT=30	İst. EP	1940	3041	3044	2714	2332
	EP	1869	1377	1270	1198	1161
	Toplam	3335	4332	4286	3902	3487
LİMİT=40	İst. EP	1637	2352	2278	1995	1690
	EP	1362	959	868	836	838
	Toplam	2658	3239	3131	2825	2525
LİMİT=50	İst. EP	1373	1844	1756	1521	1280
	EP	1083	757	701	675	663
	Toplam	2188	2550	2448	2192	1941

4.3 Veri Setlerinin Oluşturulması

Makine öğrenmesi tekniklerini uygularken kullanılacak olan veri setleri matris formatta hazırlanmıştır. Bir önceki adımda oluşturulan 1, 2, 3, 4, 5 gramlık özellikler veri setlerinin sütunlarını oluşturmaktadır. Veri setlerinin satırlarını ise işlenmemiş verilerde yer alan bağlantı metinleri oluşturmaktadır. Satırlar oluşturulurken;

- İşlenmemiş formatta bulunan veri ele alınır. e-posta numarası, bağlantı ve bağlantı metni ayrıştırılır.
- Bağlantı metni ön işleme tabi tutulur.
- Bağlantı metninin kullanılan özellik vektörüne göre gramları çıkartılır.
- Bağlantı metninin gramları ile özellik vektörü karşılaştırılır. Örtüşen özellikler için “1”, örtüşmeyen özellikler için “0” yazılır.
- Bağlantı metninde aynı gramlar yer alıyor ise bulunan gram sayısı kadar sayı hesaplanır. Veri setinde “1” yerine toplam sayı yazılır.
- Eğer bir bağlantı metninin gramları, özellik vektöründeki hiçbir özellik ile örtüşmüyor ise bu bağlantı metni veri seti kümesine dahil edilmez.
- Bağlantı metninden oluşturulan her bir özellik, özellik vektörü ile karşılaştırılırken etiket bilgisi eklenmelidir.

Yukarıdaki algoritma mantığını bir örnek ile Çizelge 4.8 ve 4.9'daki gibi gösterilebilir. Buradaki bağlantı metni veri kümesinden alınmış bir örnektir. Bağlantı metninin 3 gramları oluşturulmuştur. Veri kümesinin diğer verileri de hesaba katılarak çıkarılan özellik vektörünün bir kısmı Çizelge 4.8'deki gibidir. Oluşturulan her bir 3 gram, özellik vektörü ile tek tek karşılaştırılarak eşleşme olup olmadığı kontrol edilir. Eşleşme olması durumunda veri setindeki sayı 1 artırılır. Örneğin Çizelge 4.9'da 3 gram listesinde "REVERSE FATTY LIVER" gramı 2 defa yer aldığından veri setinde "REVERSE FATTY LIVER " özelliği "2" olarak yazılmıştır.

Çizelge 4-8 Özellik çıkartılması örneği

Bağlantı metni
REVERSE FATTY LIVER LOSE WEIGHT IMPROVE YOUR OVERALL HEALTH REVERSE FATTY LIVER

Bağlantı metninin 3 Gramları
REVERSE FATTY LIVER
FATTY LIVER LOSE
LIVER LOSE WEIGH
LOSE WEIGHT IMPROVE
WEIGHT IMPROVE YOUR
IMPROVE YOUR OVERALL
YOUR OVERALL HEALTH
OVERALL HEALTH REVERSE
HEALTH REVERSE FATTY
REVERSE FATTY LIVER

3 Gram Özellik Vektörü
YOUR OVERALL HEALTH
FATTY LIVER REMEDY
LIVER REMEDY CLICK
REMEDY CLICK HERE
CLICK HERE CLAIM

Çizelge 4-9 Örnek veri seti matrisi

Bağlantı Metni	Sınıf Etiketi	YOUR OVERALL HEALTH	FATTY LIVER REMEDY	LIVER REMEDY CLICK	REMEDY CLICK HERE	REVERSE FATTY LIVER
REVERSE FATTY LIVER						
FATTY LIVER LOSE						
LIVER LOSE WEIGH						
LOSE WEIGHT IMPROVE						
WEIGHT IMPROVE YOUR						
IMPROVE YOUR OVERALL	**İst. EP.**	**1**	**0**	**0**	**0**	**2**
YOUR OVERALL HEALTH						
OVERALL HEALTH REVERSE						
HEALTH REVERSE FATTY						
REVERSE FATTY LIVER						

Veri setini oluşturan algoritma Çizelge 4.10'daki gibidir;

Çizelge 4-10 Veri Seti Oluşturma Algoritması sözde kodları

Veri Seti OluşturmaAlgoritması

```
Girdi: F[n]=x gram özellik vektörü, n: özellik sayısı
       D: Ön işlemden geçmiş veri seti (sadece bağlantı metinlerini içerir)
       N: N gram boyutu
Çıktı: X[n]= Veri seti vektörü, n: özellik sayısı

SET X[n] "Sıfır"

foreach (verisatırı in D)
{
    Ngram Listesi=CALL NGramOluştur(VeriSatırı,N)
    Ngram sayisi=NgramListesi uzunluğu

    for(i=0 to "özellik sayısı" step 1)
    {
      for(a=0 to "Ngram sayisi" step 1)
      {
       If(NgramListesi[a]=F[i])
       X[i]+=1
      }
     }
     X[n] dosyaya yaz
     SET X[n] "Sıfır"
}
```

Veri seti matrisinin sütunları için daha önce anlatılan limit değerleri (30, 40, 50) göz önüne alınarak farklı veri setleri oluşturulmuştur. Makina öğrenmesi tekniklerinin uygulanması için kullanılacak aracın kapasitesinden dolayı en fazla 50.000 satır veri işlenebilmektedir. Bu nedenle yukarıda anlatılan algoritma mantığı ile en fazla 50000 satır olacak şekilde tüm veri setleri hazırlanmıştır. Toplam 15 farklı veri seti hazırlanmıştır. Veri setlerinin boyutları aşağıdaki gibidir;

Çizelge 4-11 Limitlere göre veri seti matris boyutları

		1 Gram	2 Gram	3 Gram	4 Gram	5Gram
30 LİMİTİ	İst. EP	25000	25000	25000	25000	23677
	EP	25000	25000	19864	8818	5125
	Toplam	50000	50000	44864	33818	28802
40 LİMİTİ	İst. EP	25000	25000	25000	25000	21592
	EP	25000	25000	18045	7613	4081
	Toplam	50000	50000	43045	32613	25673
50 LİMİTİ	İst. EP	25000	25000	25	25000	3882
	EP	25000	25000	17299	7368	19350
	Toplam	50000	50000	17324	32368	23232

	1 Gram	2 Gram	3 Gram	4 Gram	5Gram
30 LİMİTİ	50000 X 3335	50000 X 4332	44864 X 4286	33818 X 3902	28802 X 3487
40 LİMİTİ	50000 X 2658	50000 X 3239	43045 X 3131	32613 X 2825	25673 X 2525
50 LİMİTİ	50000 X 2188	50000 X 2550	17324 X 2448	32368 X 2192	23232 X 1941

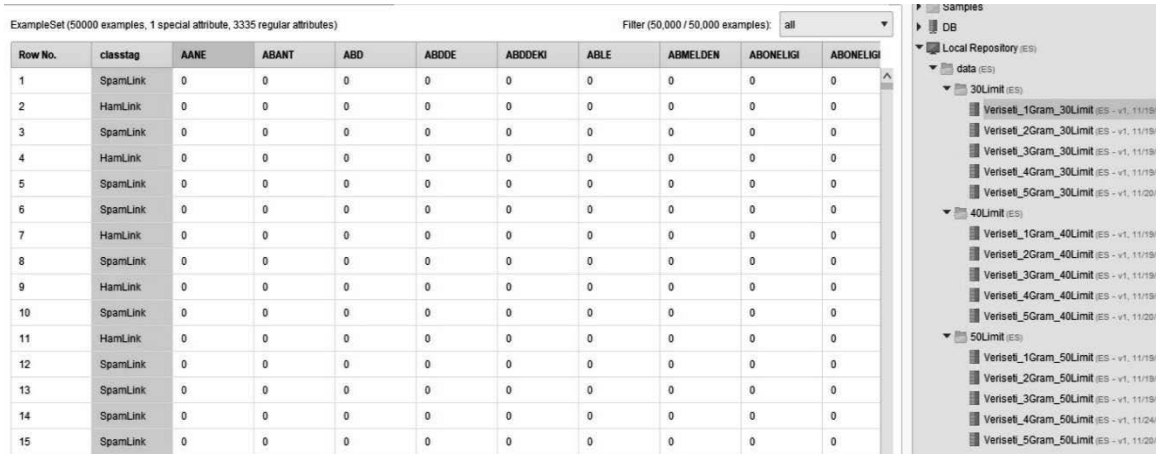
4.4 Araç Seçimi

Veri setleri oluşturulduktan sonra makine öğrenmesi tekniklerinin uygulanması için uygun araç seçimi araştırması yapılmıştır. Öncelikle makine öğrenmesi çalışmalarında en çok kullanılan ve en popüler araçlardan Weka [36] aracı ile çalışılmıştır. Weka içerisinde birçok farklı makine öğrenmesi algoritmalarını hazır olarak bulunduran bir araçtır. Weka, algoritmaları doğrudan bir veri kümesine uygulama veya kendi Java kodunuzdan çağırma imkânı tanımaktadır. Weka, veri ön işleme, sınıflandırma, kümeleme, ilişki kuralları ve görselleştirme araçları içerir. Aynı zamanda yeni makine öğrenme planları geliştirmek için de uygundur. Bu açılardan Weka aracının özellikleri bu tez çalışmasının ihtiyaçlarını karşılamak için uygun bulunmuştur. Ancak veri setleri matrislerinin boyutlarının oldukça büyük olması nedeniyle Weka [37] tarafından da raporlanmış olan "bellek aşımı" (out of memory exception) ile karşılaşılmıştır. Daha fazla RAM kapasitesine sahip bilgisayar ortamında ve WEKA'nın belirttiği çözüm yolları ile bellek aşımı hatasını çözmek için denemeler yapılmıştır. Ancak her durumda büyük boyutlu veri setleri için aynı hata ile karşılaşılırken, küçük boyutlu veri setlerinde sıkıntı yaşanmamıştır. Bu durumda başka bir araç seçimi için araştırmalar yapılmıştır. RapidMiner aracı ile çalışmaya karar verilmiştir [38]. RapidMiner Studio, veri bilimcileri için ücretsiz bir iş akışı tasarımcısıdır. Görsel tasarım olanaklarının oldukça kullanışlı olmasından dolayı fikirlerin hızlı

prototipleştirilmesine ve modellerin geçerliliğini görmeye olanak sağlar. Herhangi bir formatta tutulan verinin kolayca araç içerisine aktarılmasına olanak sağlamaktadır. Örneğin bu tez çalışmasında veriler. txt uzantılı metin dosyalarında tutulmakta idi. RapidMiner aracı ile bu veriler kolayca araç veri deposuna aktarılmış, verilerin tip, etiket vs gibi tanımlamaları yapılmıştır. RapidMiner verilerin örüntülerini kolayca tespit etmektedir. Örneğin RapidMiner ortamına aktarmış olduğumuz verilerin istenmeyen e-posta/e-posta dağılımları araç tarafından çıkartılmış, eksik değer (missing value) kontrolü yapılmıştır. RapidMiner, verileri karıştırma, model oluşturma ve modelleri geçerleme, performansı izleme imkânı tanımaktadır. Büyük boyutlu verilerle çalışma konusunda Weka'ya göre daha üstün bir araç olduğu görülmüştür. Ayrıca Weka'ya ait tüm algoritmaları kendi bünyesinde barındırmaktadır. Bu gibi nedenlerden dolayı RapidMiner aracı bu tez çalışmasında kullanılmak üzere seçilmiştir. Çalışmalar sırasında araç kullanımına yönelik hata vs. gibi herhangi bir problemle karşılaşılmadığı görülmüştür. Çalışmalarda RapidMiner 7.6 sürümü kullanılmıştır.

4.5 Deneylerin Tasarlanması

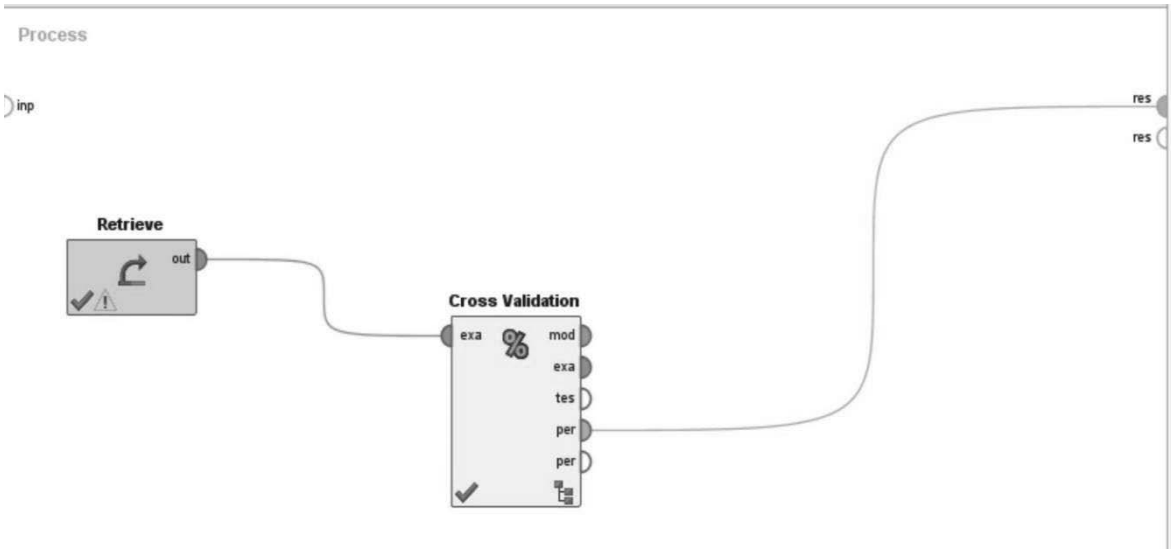
Veri setleri hazırlandıktan ve araç seçimine karar verildikten sonra son aşamada makine öğrenmesi deneyleri tasarlanmıştır. Her makine öğrenmesi metodu için 15 adet veri setine eğitim ve test işlemleri uygulanmıştır. Deneylere başlamadan önce tüm veri setleri RapidMiner ortamına yüklenmiştir. Bu işlem yapılırken "sınıf etiketi" olarak belirtilen sütun işaretlenmiştir. Şekil 4.1'de RapidMiner ortamına yüklenen veri setleri ve bu veri setlerinin görünümü gösterilmektedir.



Şekil 4-1 RapidMiner ortamında veri setinin görünümü

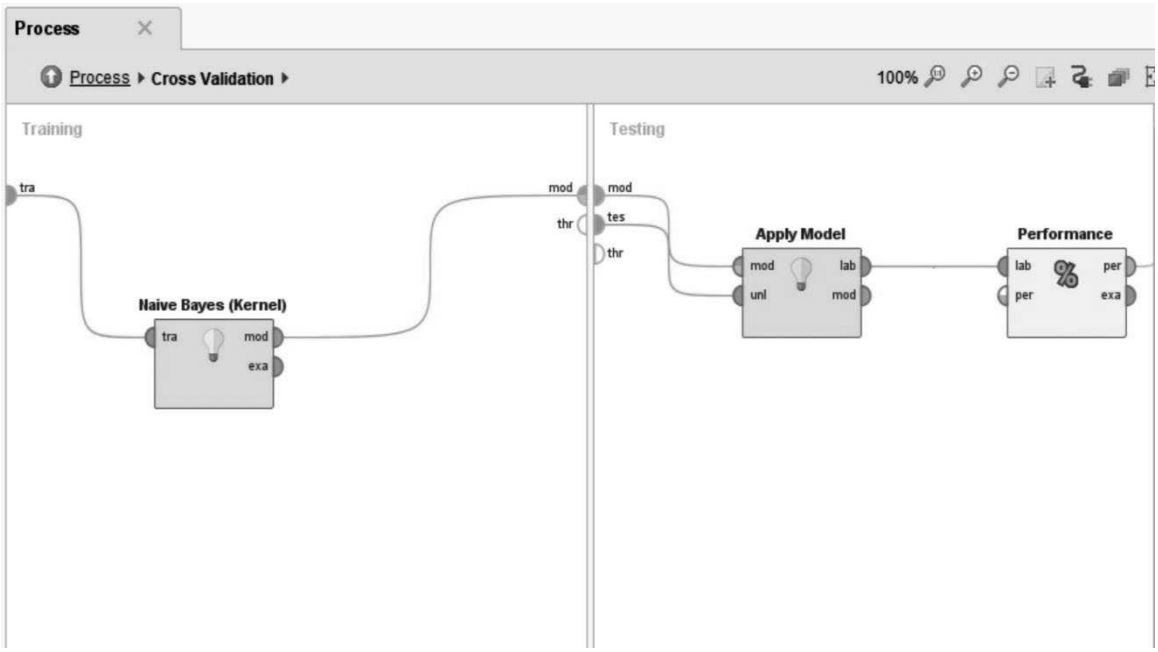
Veri setlerinin RapidMiner veri deposuna yüklenmesinin ardından ikinci aşamada deney tasarımı yapılmıştır. Tüm deneyler 10 katlamalı çapraz doğrulama (cross validation 10 folds) ile gerçekleştirilmiştir.

RapidMiner ortamında *Retrive* bloğu ile verilerin deney ortamına aktarılması sağlanmaktadır. Bu bloğun çıktısı (out) diğer blok olan *Cross Validation* bloğuna girdi sağlamaktadır. Cross Validation bloğunda ise eğitim ve test işlemleri yapılmaktadır. Cross Validation bloğunun sonucunda "per" kısaltması ile performans sonuçları çıktı olarak verilmektedir. Şekil 4.2'de RapidMiner Cross Validation gösterilmektedir.



Şekil 4-2 RapidMiner Cross Validation

Çapraz geçerleme *(cross validation)* bloğundaki eğitim ve test işlemlerini tasarlamak için bu bloğun içerisine girilir. Şekil 4.3'de Naive Bayes Kernel metodu için tasarlanmış model gösterilmektedir. Model "eğitim *(training)*" ve "test *(testing)*" olmak üzere iki bölümden oluşmaktadır. Eğitim bölümünde Naive Bayes Kernel metodu ile eğitim yapılmak istenildiği için Naive Bayes Kernel bloğu eklenmiştir. Bloğun girdisi eğitim verileri, çıktısı ise bir modeldir. Testing bölümünde ise "modelin uygulanması *(apply model)"* ve "performans *(performance)* olmak üzere iki adet blok bulunmaktadır. Model uygulanması bloğunda eğitim sonucunda ortaya çıkan model ve test verileri girdi olarak yer almaktadır. Modelin uygulanması sonucu performansın gözlemlenebilmesi için yer alan performans bloğu ise istenilen parametrelere göre model performansını göstermektedir.



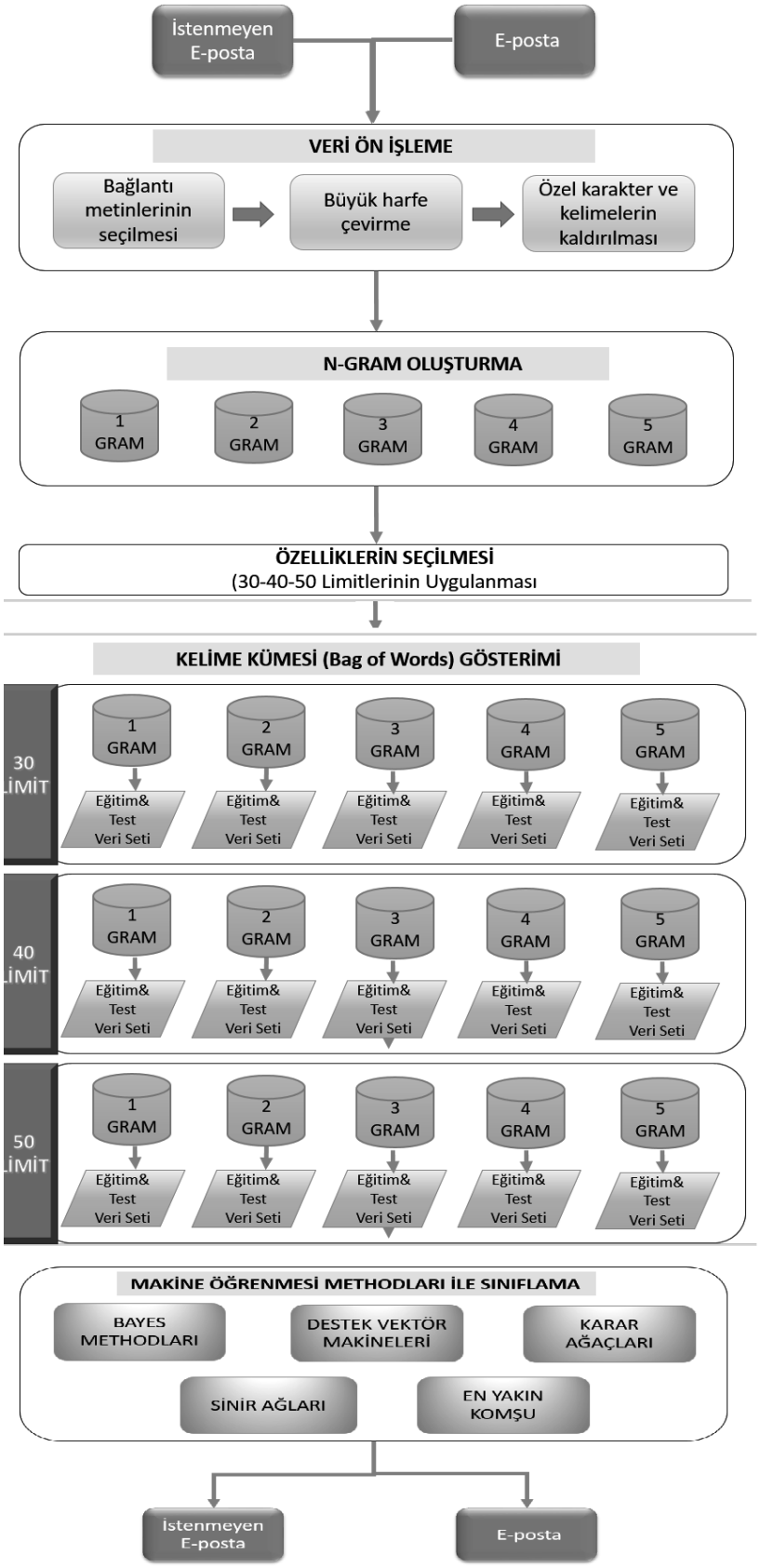
Şekil 4-3 RapidMiner ortamındaki örnek deney tasarımı

Şekil 4.3 tüm deneyler için ortaktır. Retrive bloğuna 15 adet veri seti her seferinde yeniden atanarak deneyler gerçekleştirilir. Her bir öğrenme modeli için deney tasarımı farklılaşmaktadır.

Tez çalışması kapsamında aşağıdaki makine öğrenme yöntemleri ele alınmıştır.

- Naive Bayes
- Naïve Bayes (Kernel)
- Support Vector Machine (W-SPegasos)
- Support Vector Machine (Lib SVM)
- Support Vector Machine (Linear)
- Perceptron
- KNN
- Decision Tree
- Gradient Boosted Tree
- Decision Stump
- Random Tree
- Random Forest

Buraya kadar olan aşamalar Şekil 4.4'deki gibi özetlenmiştir.



Şekil 4-4 İstenmeyen e-postaların sınıflandırılması yöntemi

5. DENEY ÇALIŞMALARI VE BULGULAR

Bu bölümde tez çalışmasında tasarlanan sürecin deneysel değerlendirmelerine ve bulgularına yer verilmiştir. Materyal ve yöntem bölümünde tanımlandığı şekilde deneysel çalışmalar yapılmıştır.

Deneysel çalışma sonuçlarının ve bulgularının anlaşılabilmesi için öncelikle performans metrikleri ele alınacaktır.

5.1 Performans Metrikleri

Bir sınıflandırma problemi için bir model oluşturulduğunda ya da var olan modeller kullanıldığında, o modelin başarısı, yapılan tüm öngörülerden gelen doğru tahminlerin sayısı gibi düşünülür. Ancak bu bilgi sadece sınıflandırmanın doğruluğunu vermektedir. Bir modelin yeterince iyi olup olmadığına karar vermek için tek başına sınıflandırma doğruluğu genellikle yeterli bilgi değildir.

Karmaşıklık matrisi (Confusion matrix): Bir sınıflandırıcının tahmin sonuçlarını sunmanın açık ve net bir yolu, bir karmaşıklık matrisi kullanmaktır. Bir karmaşıklık matrisi, gerçek değerleri bilinen bir dizi test verisi ile sınıflandırma modelinin performansını tanımlamak için sıklıkla kullanılan bir tablodur. 4 parametresi vardır [39].

Çizelge 5-1 Karmaşıklık Matrisi

		TAHMİN EDİLEN SINIF	
GERÇEK SINIF		İstenmeyen E-Posta	E-Posta
	İstenmeyen E-Posta	Doğru Pozitif(DP)	Yanlış Negatif(YP)
	E-Posta	Yanlış Pozitif(YP)	Doğru Negatif(DN)

Doğru Pozitif (DP) - Doğru tahmin edilen pozitif değerlerdir. Bu, gerçek sınıfın değeri ve tahmini sınıfın aynı olduğunu gösterir. Burada istenmeyen e-postayı, istenmeyen e-posta olarak sınıfladığımızda DP değeri bulunur.

Doğru Negatif (DN) - Bunlar doğru tahmin edilen negatif değerlerdir. Bu, gerçek sınıfın değeri ve tahmini sınıfın aynı olduğunu gösterir. Burada e-postayı, e-posta olarak sınıfladığımızda DN değeri bulunur.

Yanlış Pozitif (YP) - Bu değer gerçek sınıfınız ve tahmin edilen sınıfla çeliştiğinde ortaya çıkar. Burada bir e-postayı, istenmeyen e-posta olarak sınıfladığımızda YP değeri bulunur.

Yanlış Negatif (YN) - Bu değer gerçek sınıfınız ve tahmin edilen sınıfla çeliştiğinde ortaya çıkar. Burada bir istenmeyen e-postayı, e-posta olarak sınıfladığımızda YN değeri bulunur.

Burada doğru pozitif ve doğru negatif alanların artması istenirken, yanlış pozitif ve yanlış negatif alanların azaltılması sınıflandırma performansının iyi olduğunu göstermektedir.

Karmaşıklık matrisi sayesinde aşağıdaki metrikler hesaplanabilir.

- Doğruluk (Accuracy) : Doğruluk, en sezgisel performans ölçüsüdür ve doğru tahmin edilen gözlemin toplam gözlemlere oranıdır. Kullanılan modelde yüksek doğruluk varsa modelin en iyisi olduğunu düşünülebilir. Ancak yanlış pozitif ve yanlış negatif değerlerin sayılarının birbirinden oldukça farklı ve çok olduğu durumlarda modelin performansını değerlendirmek için diğer parametrelere bakılması gerekir [39, 40]. Doğruluk Eşitlik (9)'daki gibi hesaplanır.

$$\text{Doğruluk} = \frac{\text{DP}+\text{DN}}{\text{DP}+\text{DN}+\text{YP}+\text{YN}} \tag{9}$$

- Duyarlılık(Precision) - Duyarlılık, doğru tahmin edilen pozitif gözlemlerin, tahmin edilen toplam pozitif gözlemlere oranıdır. Buna Pozitif Öngörücü Değer(Positive Predictive Value) de denir. Duyarlılık, bir sınıflandırıcıların kesinliğinin bir ölçüsü olarak düşünülebilir. Düşük duyarlık, çok sayıda yanlış pozitifi de belirtebilir [39, 40]. Duyarlılık Eşitlik (10)'daki gibi hesaplanır.

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP}+\text{YP}} \tag{10}$$

- Hassasiyet (Recall) - Doğru tahmin edilen sonuçların, toplam pozitif sayısına oranıdır. Sınıflamadaki tüm gözlemlere yönelik olarak doğru tahmin edilen pozitif gözlemlerin oranıdır. Hassasiyet sınıflandırıcıların bütünlüğünün bir ölçüsü olarak düşünülebilir. Düşük hassasiyet, yanlış negatiflerin çok olduğunu gösterir [39, 40]. Hassasiyet Eşitlik (11)'deki gibi hesaplanır.

$$\text{Hassasiyet} = \frac{\text{DP}}{\text{DP}+\text{YN}} \tag{11}$$

- F1 skoru - Hassasiyet ve duyarlılığın harmonik ortalamasıdır. Bu nedenle, hem yanlış pozitif hem de yanlış negatifleri hesaba katar. Özellikle düzensiz bir sınıf dağılımının olduğu durumlarda F1 skoruna bakmak, doğruluk değerine bakmaktan daha yararlıdır. Yanlış pozitif ve yanlış negatiflerin benzer sayıda çıkarsa doğruluk değerine bakmak sınıflandırma başarısı için en iyi sonucu verir. Yanlış pozitiflerin ve yanlış negatiflerin sayıları çok farklıysa, hem hassasiyete hem de duyarlılığa bakmak gerekir [39, 40]. F1 skoru Eşitlik (12)'deki gibi hesaplanır.

$$\text{F1 Skoru} = 2\frac{\text{Hassasiyet}*\text{Duyarlılık}}{\text{Hassasiyet}+\text{Duyarlılık}} \tag{12}$$

5.2 Bayes Algoritmaları ile Yapılan Deneyler

5.2.1 Naive Bayes Deney Sonuçları

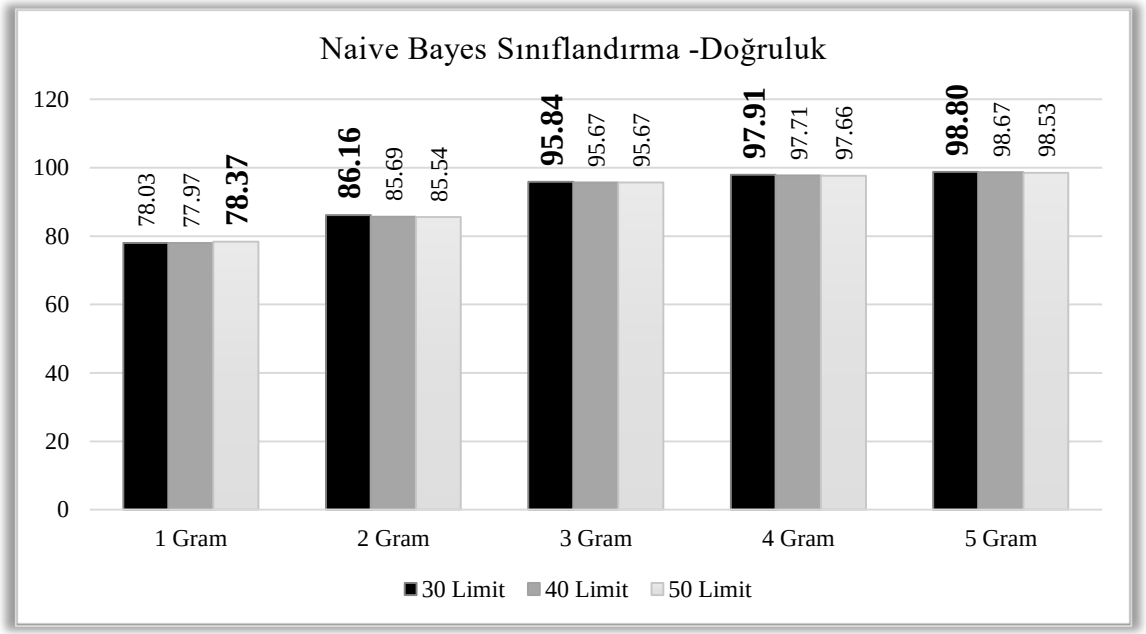
Naive Bayes Algoritması ile 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %98.8 doğruluk oranı ve %98.04 F1 Skoru elde edilmiştir. Çizelge 5.3'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır:

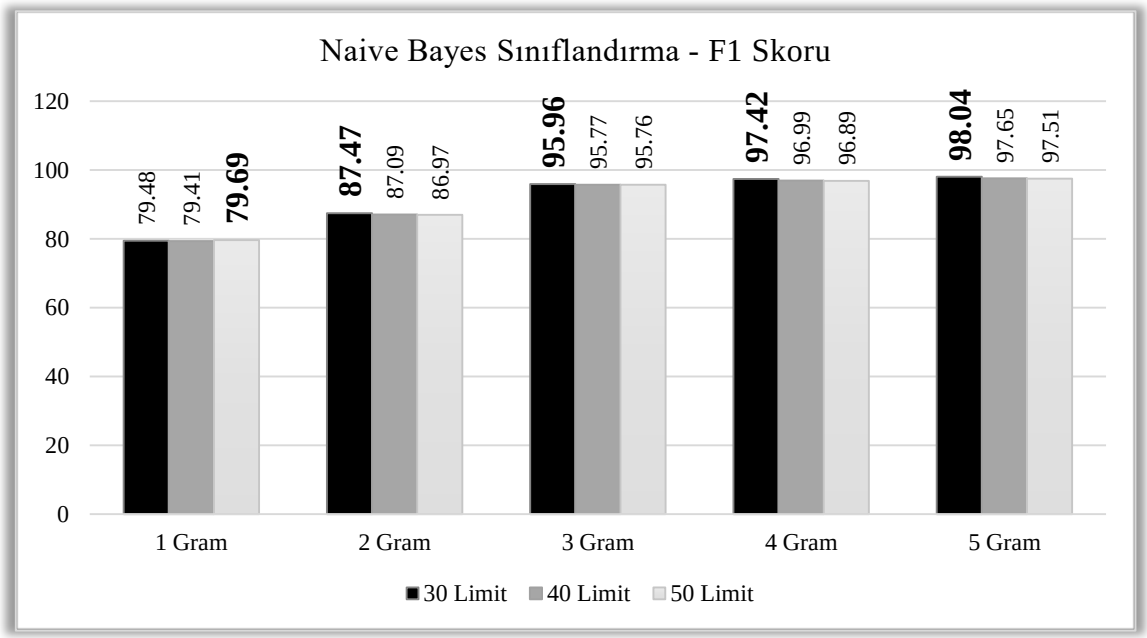
- Özellik vektöründeki kelime sayısı diğer bir deyişle gram sayısı arttıkça Naive Bayes Algoritmasının doğruluk oranı düzenli olarak artmaktadır. Bu durum özellikle metin sınıflandırmalarında uzun metinlerin sınıflandırılmasında Naive Bayes Algoritması'nın oldukça başarılı olduğunu göstermektedir.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Yapılan deneylerde 1 gram 50 limit ile yapılan bir deney hariç diğer deneylerde en yüksek doğruluğa 30 limit ile ulaşıldığı görülmüştür. Limitlere göre özellik sayılarının değişimi için Çizelge 5.2'deki verilere bakıldığında en yüksek doğruluğun elde edildiği 30 limiti ile diğer limitler arasında yaklaşık 1500 fark olduğu görülmektedir. Bu göstergeler açıkça göstermektedir ki özellik vektörünün boyutunun artmasının Naive Bayes Algoritması için olumsuz bir etki yaratmadığı aksine performansa daha iyi etki ettiği sonucuna ulaştırmıştır.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- Naive Bayes %98.8 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-2 Özellik sayıları arasındaki farklar

	1 Gram	2 Gram	3 Gram	4 Gram	5Gram
LİMİT=30	**3335**	**4332**	**4286**	**3902**	**3487**
LİMİT=40	2658	3239	3131	2825	2525
LİMİT=50	2188	2550	2448	2192	1941

Çizelge 5-3 Naive Bayes sınıflandırma deney sonuçları



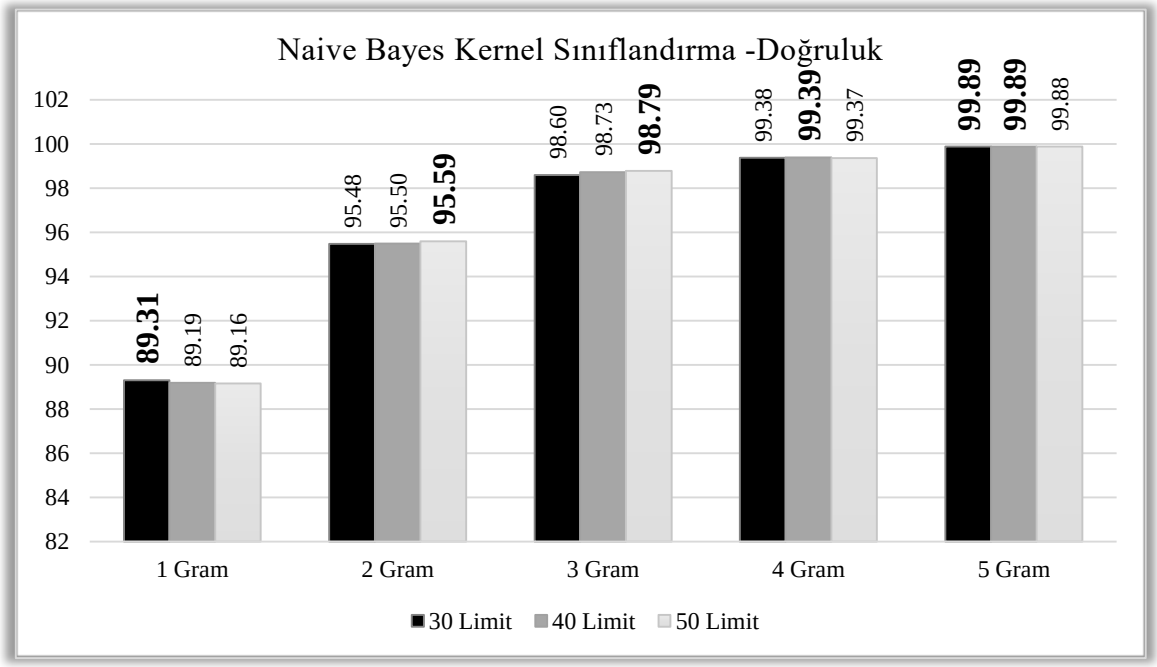


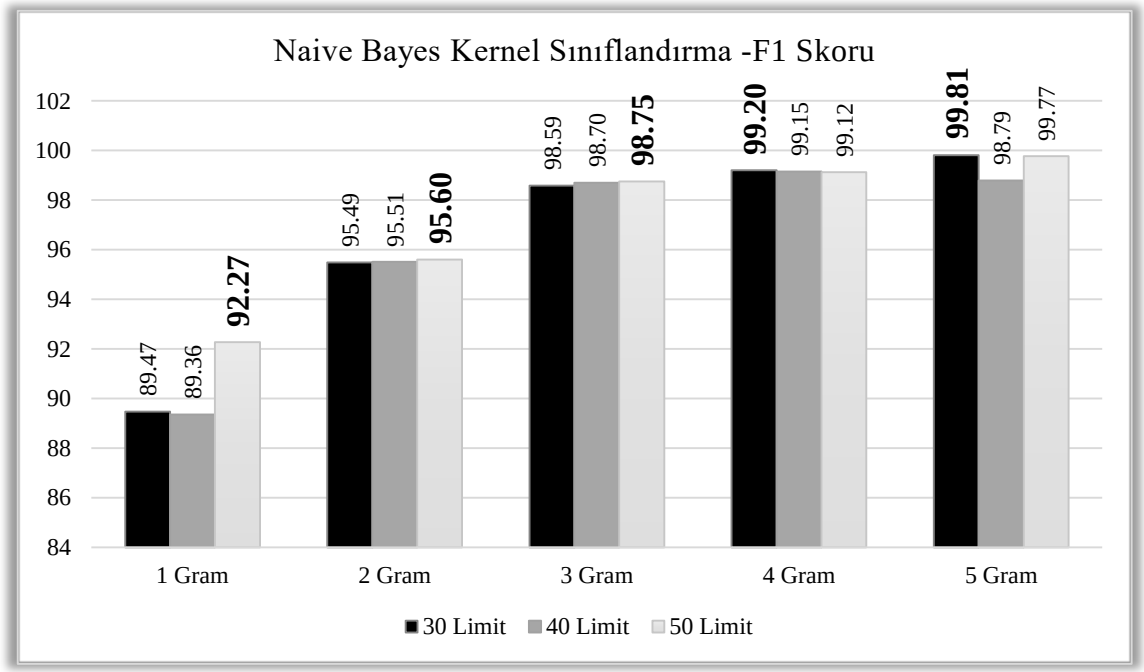
5.2.2 Naive Bayes Kernel Deney Sonuçları

Naive Bayes Kernel Algoritması ile 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %98.89 doğruluk oranı ve %99.81 F1 Skoru elde edilmiştir. Çizelge 5.4'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır.

- Özellik vektöründeki kelime sayısı diğer bir deyişle gram sayısı arttıkça Naive Bayes Kernel Algoritmasının doğruluk oranı genel olarak artmaktadır. 1 gram ve 2 gram deneyleri arasında doğruluk oranı bakımından fark fazladır. Ancak 1 gram deneylerinin F1 skorlarına baktığımızda veri dağılımında bir dengesizlik olduğunu görülmektedir. Doğruluk oranları arasındaki anormal farkın nedeni budur. 4 gram deneylerinde doğruluk oranının 3 grama göre düşüktür. Yine buradaki düşüşün nedeni için F1 skor değerlerine baktığımızda 3 ve 5 gram deneylerinin doğruluk ve F1 skor değerleri benzerlik gösterirken 4 gram deneylerinde dengesizlik olduğu görülmektedir. Buradaki sonuçlardan da görüleceği üzere f1 skor ve doğruluk değerini birlikte yorumlamak sonuçlar arasındaki anormal farklılıkların anlamak için gereklidir. Bu anormal farklılıklar önemsenmediğinde genel olarak Naive Bayes Kernel Algoritmasının tıpkı Naive Bayes Algoritması gibi artan gram sayısına olumlu bir tepki vermiştir.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan veri setleri ile yapılan deneyler incelenmiştir. Deney sonuçlarının düzenli bir dağılım göstermediği, küçük farklar göz ardı edildiğinde ise 50 limit ile sınırlandırılan özelliklerin sınıflandırma için daha iyi olduğu görülmüştür. Naive Bayes'in aksine Naive Bayes Kernel algoritmasının özellik sayısının artmasına karşı olumsuz etki gösterdiği gözlemlenmiştir.
- Doğruluk ve F1 Skoru karşılaştırıldığında 1 gram ve 4 gram ile hazırlanan veri setleri ile yapılan deneylerde limitlere göre dengesizlik olduğu ve bu durumun performansa olumsuz etki ettiği görülmüştür.
- Naive Bayes Kernel, %98.89 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında başarılı bulunmuştur.

Çizelge 5-4 Naive Bayes Kernel sınıflandırma deney sonuçları





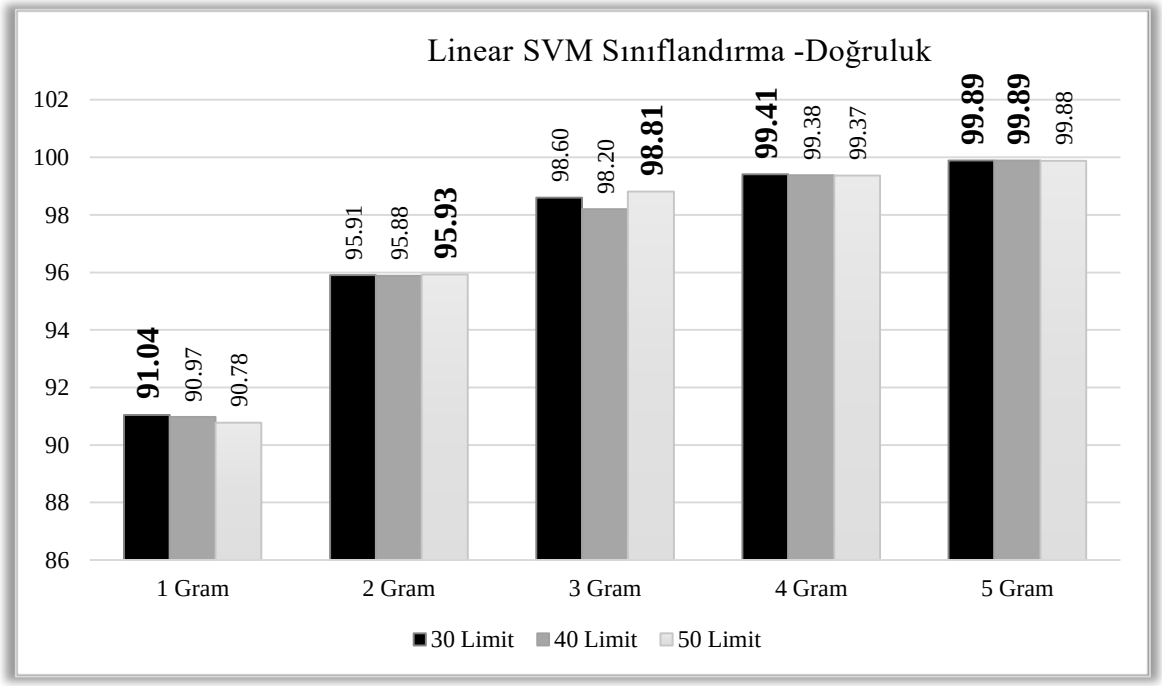
5.3 Destek Vektör Makinaları Algoritmaları ile Yapılan Deneyler

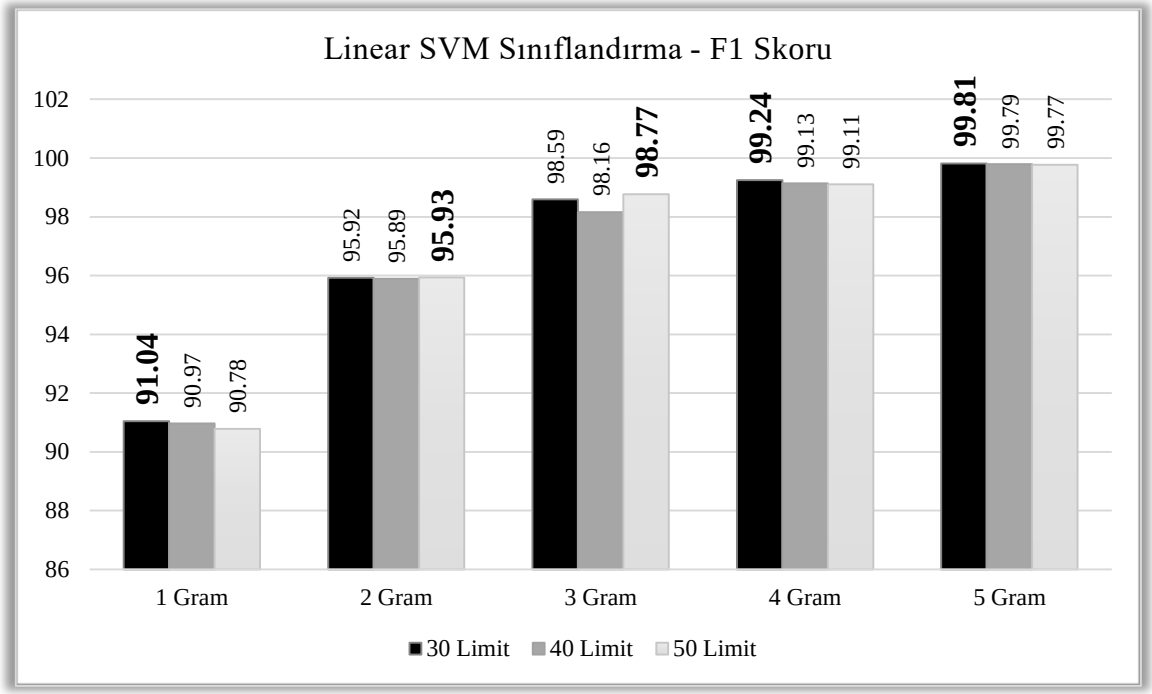
5.3.1 Doğrusal SVM (Linear SVM) Deney Sonuçları

Linear SVM Algoritması ile 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %99.89 doğruluk oranı ve %98.81 F1 Skoru elde edilmiştir. Çizelge 5.5'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır:

- Özellik vektöründeki gram sayısı arttıkça Linear SVM Algoritmasının doğruluk oranı düzenli olarak artmakta, %99,89 gibi çok yüksek bir başarı oranına ulaşmaktadır. Bu durum Naive Bayes Algoritma'ları gibi Linear SVM algoritmasının da uzun metinlerin sınıflandırılmasında oldukça başarılı olduğunu göstermektedir.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Yapılan deneylerde düzenli bir dağılım görülmemektedir. Özellik Vektörü boyutunun performansa olan etkisi değişkendir.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- Linear SVM %99.89 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-5 Linear SVM sınıflandırma deney sonuçları



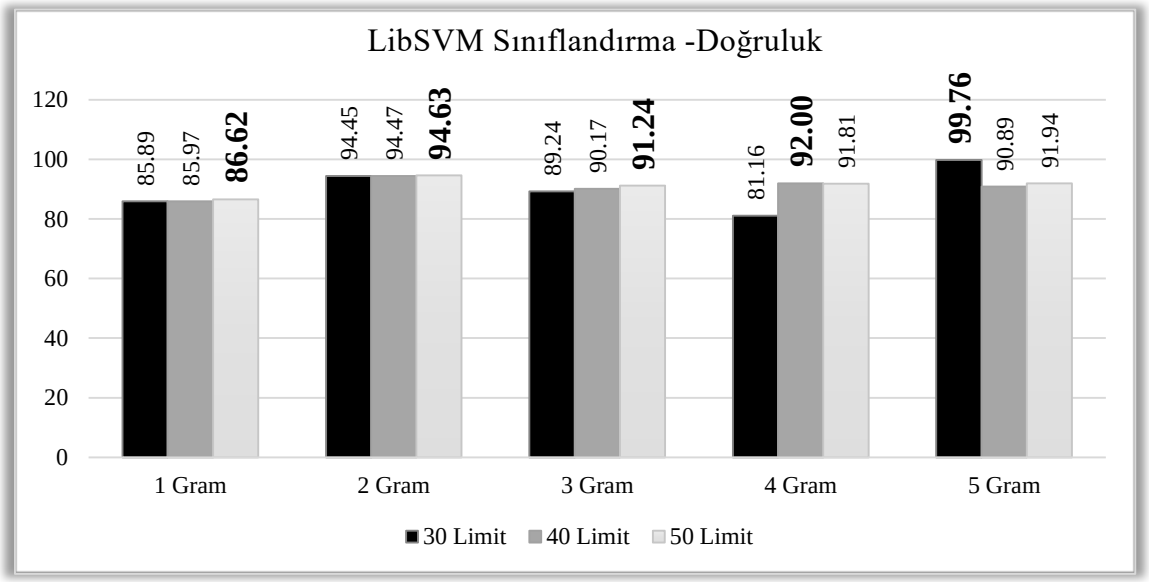


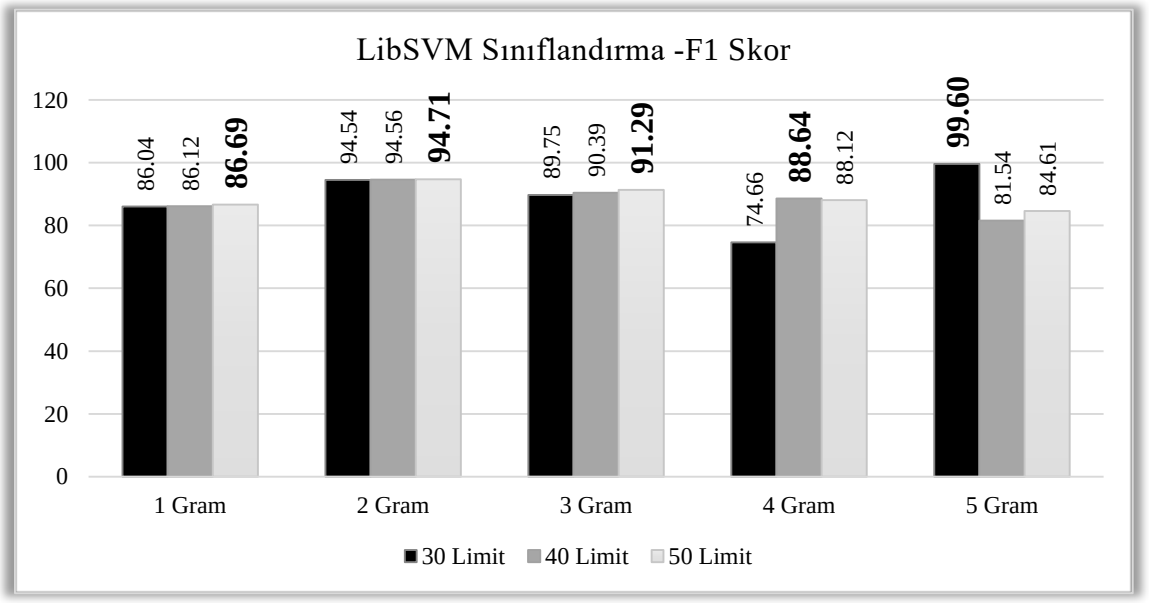
5.3.2 LibSVM Deney Sonuçları

LibSVM Algoritması ile 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %99.76 doğruluk oranı ve %99.6 F1 Skoru elde edilmiştir. Çizelge 5.6'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır.

- Özellik vektöründeki gram sayısının artması performans artışı için düzenli bir iyileştirme sağlamamıştır. 2 gram ile yapılan deneylerde performans 1 gram deneylerine göre artmışken, 3 Gram ile yapılan deneylerde doğruluk oranı 2 gram deneylerine göre düşmüştür.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Yapılan deneylerde büyük çoğunlukla 50 limitinin yüksek doğruluk sağladığı görülse de en yüksek doğruluk oranına 30 limit 5 gram ile yapılan deneyle ulaşılmıştır.
- LibSVM 1 gram ile yapılmış deneylerde başarı oranı oldukça düşük çıkmıştır.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- Lib SVM, en yüksek başarısı olan %99.76 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-6 LibSVM sınıflandırma deney sonuçları





5.3.3 Pegasos SVM Deney Sonuçları

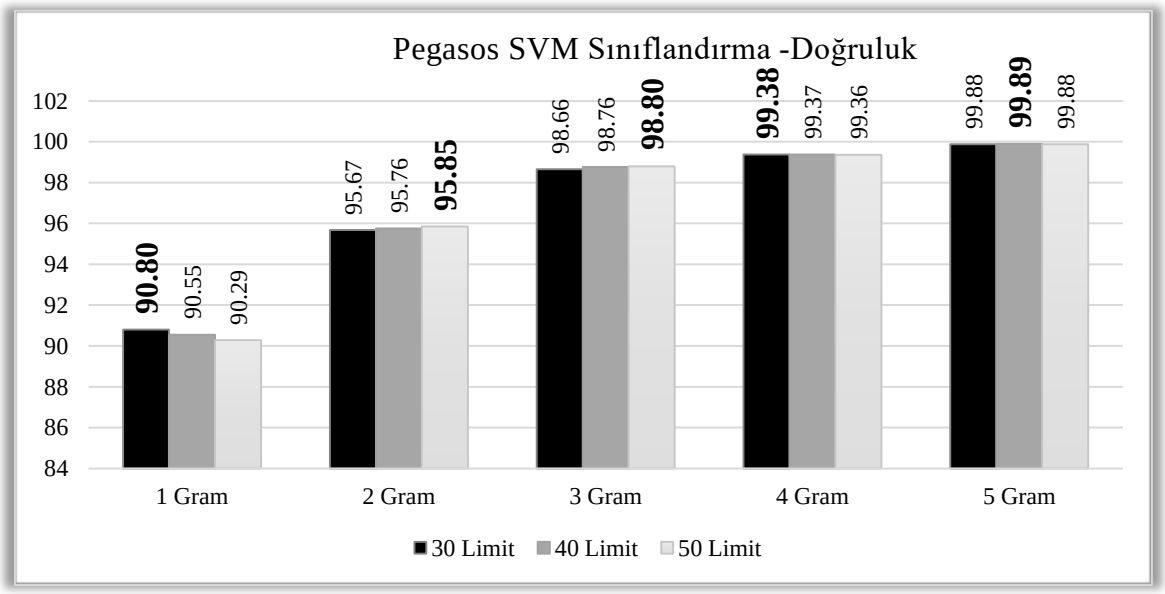
Pegasos SVM Algoritması ile 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %99.89 doğruluk oranı ve %99.8 F1 Skoru elde edilmiştir. Çizelge 5.7'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır.

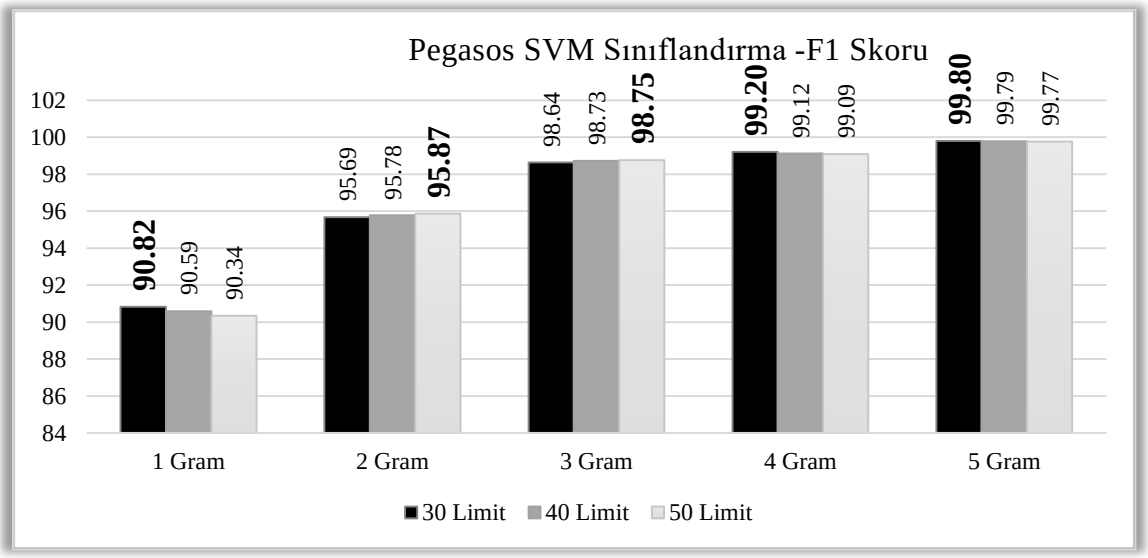
- Özellik vektöründeki kelime sayısı diğer bir deyişle gram sayısı arttıkça doğruluk oranı düzenli olarak artmaktadır. Bu durum özellikle metin sınıflandırmalarında uzun metinlerin sınıflandırılmasında Pegasos SVM Algoritması'nın oldukça başarılı olduğunu

göstermektedir. Ayrıca gram sayısı etkisi LibSVM algoritmasına göre daha net bir şekilde gözükmektedir.

- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Düzenli bir artış gözükmese de limitlerin doğruluk oranı birbirine oldukça yakındır. Özellikle 40 limit ile yapılan deneyler genel olarak başarılı gözükmektedir.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- 1 gram ile yapılan deneyler hariç genel olarak Pegasos SVM'in, Lib SVM'e göre daha iyi, Linear SVM'e ise doğruluk oranı bakımından yakın olduğu görülmüştür. Farklı gram ve limitlere göre genel olarak başarı oranı %96'nın üzerinde kalmıştır. Algoritma en yüksek başarısı olan %99.89 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-7 Pegasos SVM sınıflandırma deney sonuçları





5.4 Karar Ağaçları Algoritmaları ile Yapılan Deneyler

Gradient Boosted Trees, Decision Stump, Ramdom Tree, Random Forest gibi algoritmalarla yapılan deneylerde öncelikle 50 limit ile sınırlandırılmış 3 gram veri seti ile deneyler yapılmıştır. Alanyazın çalışmalarından elde edilen sonuçlar ve buradaki deney ortamındaki başarı oranının oldukça düşük çıkmış olması nedeniyle aslında karar ağacı algoritmalarının istenmeyen e-posta sınıflanmasına uygun olmadığı sonucuna varılmıştır. Bu sonuca varılmasındaki temel nedenler aşağıdaki gibi sıralanabilir:

- Deneylerde kullanılan veri setindeki özellik vektörünün boyutu 2448'dir. Özellik sayısının fazla olması hem ağaç oluşturma hem de ağaç budama karmaşıklığını artırmaktadır.
- Veri sayısının fazla olması ve Özellik Vektörü'nün boyutunun oldukça büyük olmasından dolayı bütün verinin özelliklerini modelleyemeyen bir ağaç oluşturulmuştur. Karar ağacı öğrenenleri, bazı sınıfları önceden öğrenmiş ise önyargılı ağaçlar oluştururlar.

Çizelge 5.8'deki sonuçlar incelendiğinde karar ağacı algoritmalarının istenmeyen e-posta sınıflandırmasında oldukça düşük başarılar elde ettiği görülmektedir.

Çizelge 5-8 Karar Ağaçları Algoritmaları ile sınıflandırma deney sonuçları

	Karar Ağaçları	**Gradient Boosted Trees**	**Decision Stump**	**Random Tree**	**Random Forest**
Doğruluk	86.49	79.19	63.26	59.10	59.41
F1- Skoru	91.00	78.78	65.52	37.15	61.71
Sınıflama Hatası	13.51% +/- 0.38%	20.81% +/- 19.15%	36.74% +/- 0.22%	40.90% +/- 0.01%	40.59% +/- 0.36%

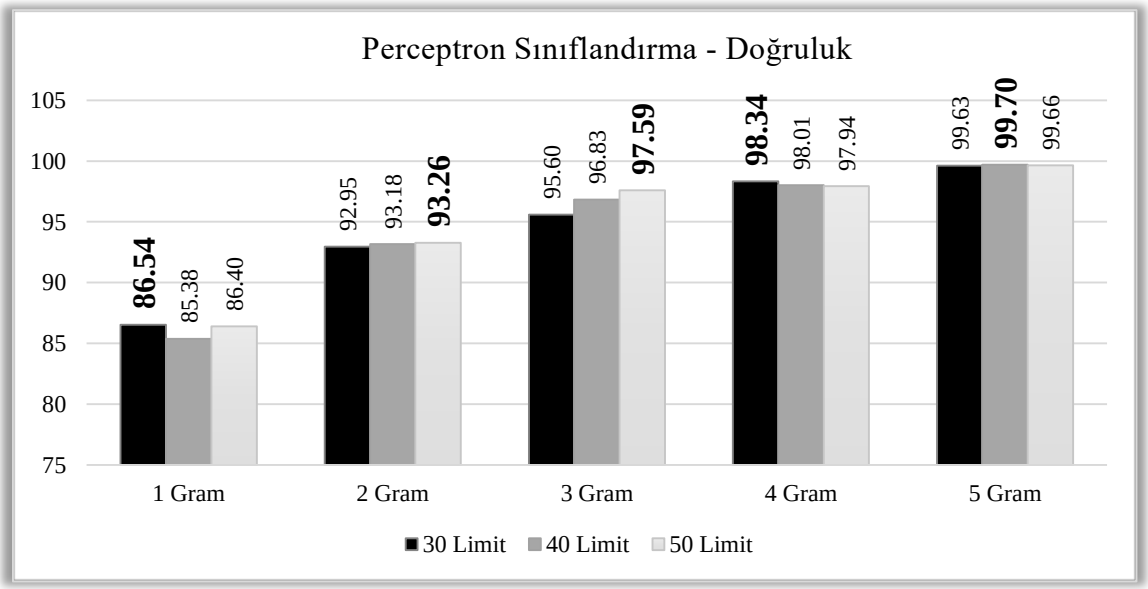
5.5 Yapay Sinir Ağları ile Yapılan Deneyler

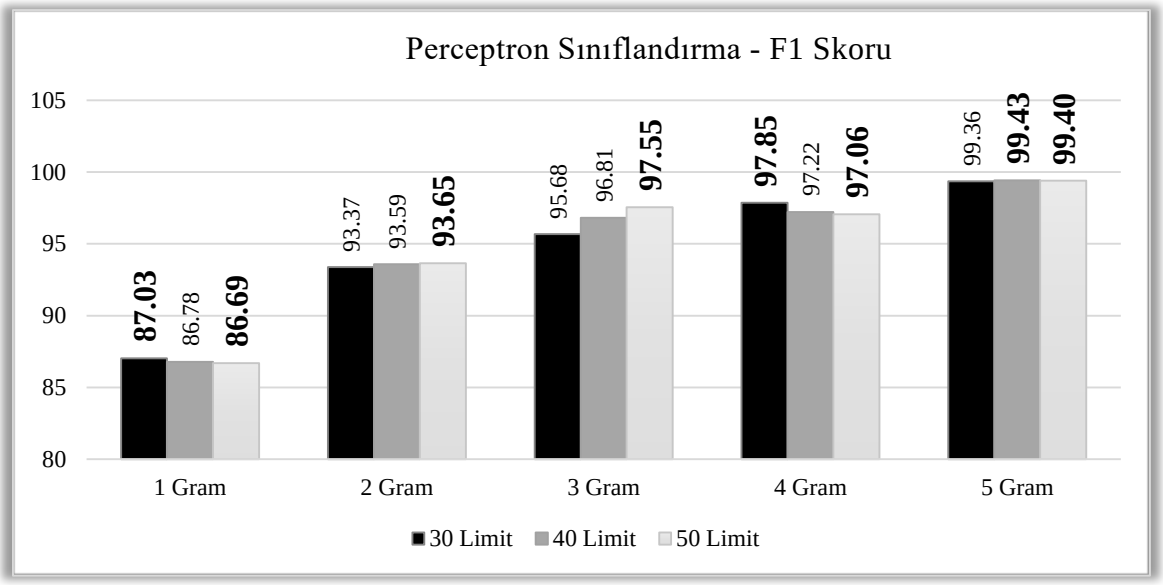
5.5.1 Perceptron Deney Sonuçları

Perceptron Algoritması ile RapidMiner ortamında1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %99.70 doğruluk oranı ve %99.43 F1 Skoru elde edilmiştir. Çizelge 5.9'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır.

- Özellik vektöründeki gram sayısı arttıkça Perceptron Algoritmasının doğruluk oranı düzenli olarak artmaktadır.
- En iyi performans sonucuna 5 gram özellik vektörü ile hazırlanan veri setlerinde olduğu görülmüştür. En düşük performansın ise 1 gram özellik vektörü ile hazırlanan veri setlerinde olduğu görülmüştür.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Yapılan deneylerde genel olarak en yüksek doğruluğa 50 limit ile ulaşıldığı görülmüştür.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- Perceptron, %98.8 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-9 Perceptron sınıflandırma deney sonuçları



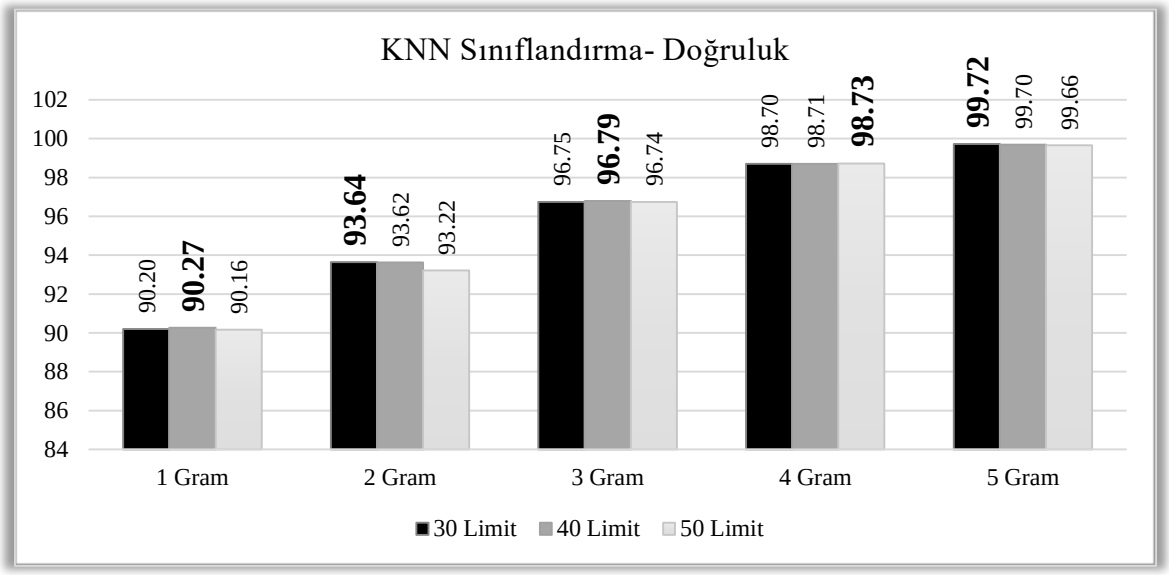


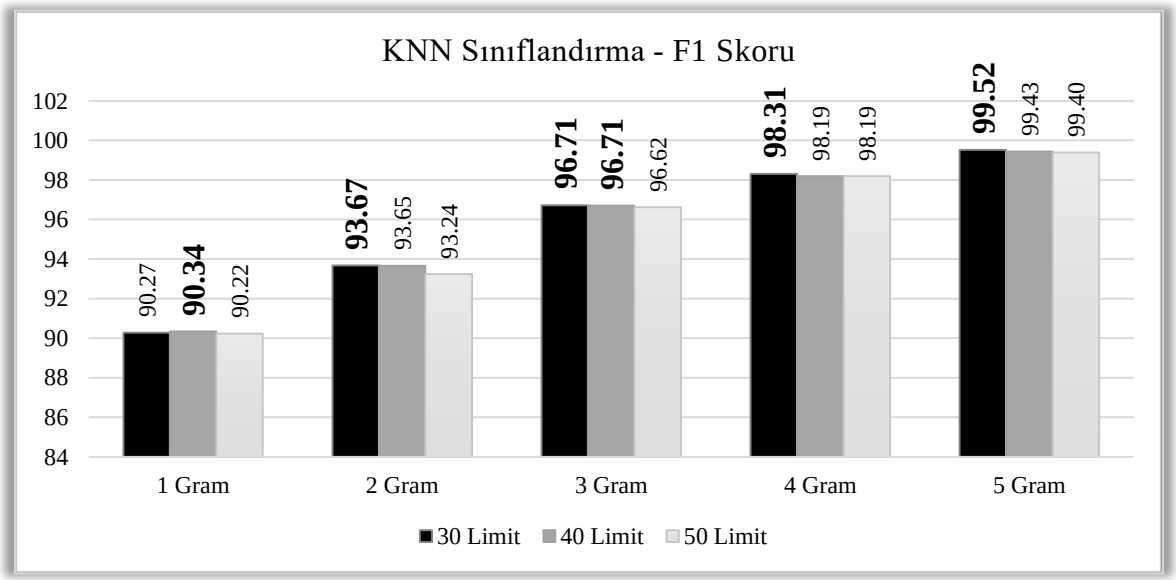
5.6 En Yakın Komşu (K-NN) Algoritması ile Yapılan Deneyler

KNN Algoritması ile RapidMiner ortamında 1,2,3,4,5 gramlık veri setleri ile yapılan deneylerde en yüksek %99.72 doğruluk oranı ve %99.52 F1 Skoru elde edilmiştir. Çizelge 5.10'de verilen doğruluk ve F1 Skoru sonuçlarına bakıldığında aşağıdaki bulgulara ulaşılmıştır:

- Özellik vektöründeki kelime sayısı diğer bir deyişle gram sayısı arttıkça KNN Algoritmasının doğruluk oranı düzenli olarak artmaktadır. Bu durum özellikle metin sınıflandırmalarında uzun metinlerin sınıflandırılmasında KNN Algoritması'nın oldukça başarılı olduğunu göstermektedir.
- Özellik sayısının performansa etkisi incelenmek istenildiğinde 30, 40, 50 limitleri ile hazırlanan deney setleri incelenmiştir. Genel olarak 30 limiti ile yapılan deneylerin yüksek doğruluk oranında olduğu görülmektedir.
- 5 gram ile yapılan deneylerin en başarılı deneyler olduğu görülmektedir.
- Doğruluk ve F1 Skoru karşılaştırıldığında sonuçların büyük ölçüde örtüştüğü görülmektedir. Bu durum veri dağılımının düzenli olduğunu göstermektedir.
- Naive Bayes %99.72 doğruluk oranı ile istenmeyen e-posta/e-posta sınıflandırmasında oldukça başarılı bulunmuştur.

Çizelge 5-10 KNN sınıflandırma deney sonuçları





6. SONUÇLAR

Bu tez çalışmasının temel amacı e-postaların içeriklerinde yer alan bağlantı metinleri ile istenmeyen e-postaların ayırt edilmesini sağlamaktır. Çalışmada kullanılacak veri setleri çeşitli önişlemlerden geçirilerek makine öğrenmesi tekniklerinde kullanılmaya hazır hale getirilmiştir. Veri setleri hazırlanırken Kelime Kümesi Tekniği'nden (BOW) yararlanılarak 1,2,3,4,5 gram olarak hazırlanan 5 adet özellik vektörü kullanılmıştır. Farklı gramlarda özellik vektörü oluşturulmasındaki temel amaç gram sayısının sınıflandırma performansına etkisini gözlemleyebilmektir. Oluşturulan N gramların sayısının çok fazla olması diğer bir deyişle özellik vektörünün boyutunun çok uzun olmasından ve bu durumu makine öğrenmesi eğitim ve test işlemlerini zorlaştırılmasından dolayı çeşitli limitler ile özellik vektörlerinin boyutları azaltılmıştır. Bu limitler 30,40,50'dir. Limitler aynı N gramların tekrar sayıları göz önüne alınarak oluşturulmuştur.

Makine öğrenmesi teknikleri uygulanırken ilk olarak 50 limit 3 Gram özellik vektörü ile hazırlanmış veri seti eğitilmiş ve test edilmiştir. Bunun sonucunda sınıflandırma doğruluk oranı ortalama %95'in üzerinde çıkan sınıflandırma algoritmaları için N gram sayısının performansa olan etkisi incelenmiştir. %95 doğruluk oranı altında kalan sınıflandırma algoritmaları için ise doğruluk oranı istenen düzeyde olmadığı için N gram sayısının performansa etkisinin gözlemlenmesi anlamlı olmamıştır.

Çalışmadaki bir diğer amaç farklı makine öğrenme tekniklerinin istenmeyen e-posta sınıflandırması için başarı düzeyini incelemektir. Analiz yapılırken öncelikle veri tipine uygun sınıflandırma algoritmaları gözden geçirilmiş, uygun bulunan algoritmalar için makine öğrenmesi deneyleri yapılmıştır. Çalışma kapsamında ele alınan makine öğrenmesi yöntemleri "Bayes, Destek Vektör Makineleri, Karar Ağaçları, Yapay Sinir Ağları ve Perceptron Algoritmaları" olarak gruplandırılmıştır. Bu gruplardan Karar Ağacı Algoritmaları hariç diğer tüm algoritmalar için N gramların performansa etkisi incelenmiştir. Karar Ağaçları Algoritmaları için ortalama doğruluk oranı ortalama %65 çıkmıştır. Tüm deneyler RapidMiner aracı kullanılarak 10 katlamalı çapraz doğrulama (cross validation 10 folds) ile gerçekleştirilmiştir.

Tüm deneyler bittikten sonra analiz işlemleri için doğruluk, F1 skoru ve sınıflama hata oranı metrikleri ile istenmeyen e-posta sınıflandırma performansı incelenmiştir.

Çizelge 6.1'de yer alan sınıflandırma algoritmalarının doğruluk oranları incelendiğinde şu sonuçlara ulaşılmıştır:

- Gram sayılarına göre bakıldığında tüm deneylerde en düşük doğruluk oranının 1 gram özellik vektörü ile hazırlanmış olan veri setinden elde edildiği görülmektedir. Bu durum istenmeyen e-postaların sınıflandırılmasında 1 gram özellik vektörü ile hazırlanmış olan veri setlerinin istenmeyen e-postaların ayırt edilmesi için yetersiz olduğu sonucuna ulaştırmıştır.
- Yine gram sayılarına göre bakıldığında N gram sayısı arttıkça makine öğrenme algoritmalarının doğruluk oranlarının da kademeli olarak arttığı görülmektedir. Bu durum N gram ile yapılan istenmeyen e-posta sınıflandırma çalışmalarında N gramdaki N sayısının artmasının sınıflandırma doğruluk oranı için oldukça iyi bir çözüm olacağını göstermektedir.
- Algoritmaların büyük çoğunluğunda en yüksek doğruluk oranına 5 gram 30 limit veri seti ile yapılmış deneylerde ulaşıldığı görülmüştür. Diğer veri setleri ile yapılan deneylerde de yine en yüksek doğruluğu 5 gram 40 Limit veri seti ile yapılan deneylerde ulaşılmıştır. Bu durum 5 gram özellik vektörü ile hazırlanan veri setlerinin istenmeyen e-postaların sınıflandırılmasında önemli bir fark yarattığını göstermektedir.
- Algoritmaların istenmeyen E-postaların sınıflandırılmasındaki genel başarısına bakıldığında Karar Ağaçları Algoritma'larından Decision Tree, Gradient Boosted Trees, Decision Stump, Random Tree ve Random Forest Algoritmalarının oldukça düşük doğruluk oranı ile sınıflandırma yapmıştır. Bunun tersine Çizelge 6.1'de verilen sonuçlara göre Naive Bayes (%98.8), Naive Bayes Kernel (%99.89), LibSVM (%99.76), LinearSVM (%99.89), Pegasos SVM (%99.89), Perceptron (%99.7), KNN(%99.72) algoritmaları ise %98.8'in üzerinde doğruluk oranı istenmeyen E-posta sınıflandırılmasında oldukça başarılı bulunmuştur.

Çizelge 6-1 Sınıflandırma Algoritmaları ile yapılan deneylerin doğruluk sonuçları

Veri seti		Methodların Doğrulukları						
	N Gram	NB	NBK	Lib SVM	Linear SVM	Pegasos	Percep.	K-NN
Limit=30	1 Gram	78.03	89.31	85.89	91.04	90.80	86.54	90.20
	2 Gram	86.16	95.48	94.45	95.91	95.67	92.95	93.64
	3 Gram	95.84	98.60	89.24	98.60	98.66	95.60	96.75
	4 Gram	97.91	99.38	81.16	99.41	99.38	98.34	98.70
	5 Gram	**98.80**	**99.89**	**99.76**	**99.89**	99.88	99.63	**99.72**
Limit=40	1 Gram	77.97	89.19	85.97	90.97	90.55	85.38	90.27
	2 Gram	85.69	95.50	94.47	95.88	95.76	93.18	93.62
	3 Gram	95.67	98.73	90.17	98.20	98.76	96.83	96.79
	4 Gram	97.71	99.39	92.00	99.38	99.37	98.01	98.71
	5 Gram	98.67	**99.89**	90.89	**99.89**	**99.89**	**99.70**	99.70
Limit=50	1 Gram	78.37	89.16	86.62	90.78	90.29	86.40	90.16
	2 Gram	85.54	95.59	94.63	95.93	95.85	93.26	93.22
	3 Gram	95.67	98.79	91.24	98.81	98.80	97.59	96.74
	4 Gram	97.66	99.37	91.81	99.37	99.36	97.94	98.73
	5 Gram	98.53	99.88	91.94	99.88	99.88	99.66	99.66

- Özellik vektörü boyutuna getirilen limitlere göre doğruluk oranı için Çizelge 6.2'deki değerlere bakıldığında özellik vektörü boyutundan en az etkilenen algoritmalar: Naive Bayes Kernel, Linear SVM, Pegasos SVM, KNN olmuştur. En çok etkilenen ise 4 gram özellik vektörü için Lib SVM olmuştur.

Çizelge 6-2 Özellik vektörü limitlerine göre doğruluk oranları arasındaki farklar

	Limitlere Göre Doğruluk Oranları Arasındaki Farklar							
	N Gram	**NB**	**NBK**	**Lib SVM**	**Linear SVM**	**Pegasos**	**Perceptron**	**K-NN**
30 Limit-40 Limit Farkı	1 Gram	0.06	0.12	0.08	0.07	0.25	1.16	0.07
	2 Gram	0.47	0.02	0.02	0.03	0.09	0.23	0.02
	3 Gram	0.17	0.13	0.93	**0.40**	0.10	1.23	0.04
	4 Gram	0.20	0.01	**10.84**	0.03	0.01	0.33	0.01
	5 Gram	0.13	0.00	8.87	0.00	0.01	0.07	0.02
40 Limit-50 Limit Farkı	1 Gram	0.40	0.03	0.65	0.19	0.26	1.02	0.11
	2 Gram	0.15	0.09	0.16	0.05	0.09	0.08	0.40
	3 Gram	0.00	0.06	1.07	0.61	0.04	0.76	0.05
	4 Gram	0.05	0.02	0.19	0.01	0.01	0.07	0.02
	5 Gram	0.14	0.01	1.05	0.01	0.01	0.04	0.04
30 Limit-50 Limit Farkı	1 Gram	0.34	0.15	0.73	0.26	**0.51**	0.14	0.04
	2 Gram	**0.62**	0.11	0.18	0.02	0.18	0.31	**0.42**
	3 Gram	0.17	**0.19**	2.00	0.21	0.14	**1.99**	0.01
	4 Gram	0.25	0.01	10.65	0.04	0.02	0.40	0.03
	5 Gram	0.27	0.01	7.82	0.01	0.00	0.03	0.06

- Çizelge 6.3'deki sınıflandırma hata oranlarına bakıldığında Çizelge 6.1'deki sonuçları desteklediği görülmektedir. Tüm algoritmalarda en yüksek sınıflandırma hatasının 1 gram özellik vektörü ile hazırlanmış veri setlerinde olduğu görülmektedir.

Çizelge 6-3 Sınıflandırma hata oranları

Veri seti		Methodların Sınıflama Hata Oranı						
	N Gram	NB	NBK	Lib SVM	Linear SVM	Pegasos	Percep.	K-NN
Limit=30	1 Gram	21.97% +/- 0.41%	**10.69% +/- 0.47%**	**14.11% +/- 0.44%**	8.96% +/- 0.29%	4.33% +/- 0.22%	13.46% +/- 0.42%	9.84% +/- 0.43%
	2 Gram	13.84% +/- 0.56%	4.52% +/- 0.32%	5.55% +/- 0.28%	4.09% +/- 0.21%	1.34% +/- 0.18%	7.05% +/- 0.32%	6.78% +/- 0.42%
	3 Gram	4.16% +/- 0.31%	1.40% +/- 0.17%	10.76% +/- 0.86%	1.40% +/- 0.16%	1.34% +/- 0.18%	4.40% +/- 1.51%	3.26% +/- 0.29%
	4 Gram	2.09% +/- 0.27%	0.62% +/- 0.14%	18.84% +/- 0.48%	0.59% +/- 0.12%	0.62% +/- 0.11%	1.66% +/- 0.52%	1.27% +/- 0.34%
	5 Gram	1.20% +/- 0.20%	0.11% +/- 0.07%	0.24% +/- 0.08%	0.11% +/- 0.07%	0.12% +/- 0.07%	0.37% +/- 0.07%	0.34% +/- 0.13%
Limit=40	1 Gram	**22.03% +/- 0.50%**	10.81% +/- 0.28%	14.03% +/- 0.55%	**9.03% +/- 0.30%**	9.45% +/- 0.26%	**14.62% +/- 0.48%**	9.80% +/- 0.67%
	2 Gram	14.31% +/- 0.32%	4.50% +/- 0.23%	5.53% +/- 0.34%	4.12% +/- 0.25%	4.24% +/- 0.29%	6.82% +/- 0.29%	6.36% +/- 0.49%
	3 Gram	4.33% +/- 0.27%	1.27% +/- 0.12%	9.83% +/- 0.15%	1.80% +/- 1.72%	1.24% +/- 0.09%	3.17% +/- 0.35%	3.25% +/- 0.32%
	4 Gram	2.29% +/- 0.23%	0.61% +/- 0.14%	8.00% +/- 0.21%	0.62% +/- 0.15%	0.63% +/- 0.14%	1.99% +/- 0.30%	1.30% +/- 0.17%
	5 Gram	1.33% +/- 0.21%	0.11% +/- 0.07%	9.11% +/- 0.46%	0.11% +/- 0.07%	0.11% +/- 0.07%	0.30% +/- 0.12%	0.28% +/- 0.09%
Limit=50	1 Gram	21.63% +/- 0.31%	10.84% +/- 0.25%	13.38% +/- 0.52%	9.22% +/- 0.47%	**9.71% +/- 0.43%**	13.60% +/- 0.67%	**9.73% +/- 0.41%**
	2 Gram	14.46% +/- 0.34%	4.41% +/- 0.26%	5.37% +/- 0.18%	4.07% +/- 0.28%	4.15% +/- 0.24%	6.74% +/- 0.27%	6.38% +/- 0.40%
	3 Gram	14.46% +/- 0.34%	1.21% +/- 0.11%	8.76% +/- 0.41%	1.19% +/- 0.12%	1.20% +/- 0.13%	2.41% +/- 0.20%	3.21% +/- 0.26%
	4 Gram	2.34% +/- 0.27%	0.63% +/- 0.10%	8.19% +/- 0.76%	0.63% +/- 0.10%	0.64% +/- 0.10%	2.06% +/- 0.28%	1.29% +/- 0.18%
	5 Gram	1.47% +/- 0.22%	0.12% +/- 0.08%	8.06% +/- 0.27%	0.12% +/- 0.08%	0.12% +/- 0.08%	0.34% +/- 0.13%	0.30% +/- 0.12%

Sonuç olarak bu tez çalışmasında içerisinde bağlantı ve bu bağlantılarla ilişkili metin barındıran e-postalara karşı makine öğrenme yöntemleri incelenmiş, doğruluk oranı oldukça yüksek olan sonuçlar elde edilmiştir. Çalışma içerisinde bağlantı barındırmayan e-postaları kapsamamaktadır. Bu nedenle gelecekteki çalışmalarda buradaki çalışmaya ek olarak gövde

(body), başlık (header) gibi alanlar da dâhil edilerek istenmeyen e-postaların tespit edilmesi amaçlanmaktadır. Yapılan çalışma aynı zamanda alanyazında yer alan çalışmalarda elde edilen sonuçları destekleyici nitelikte olmuştur.

KAYNAKLAR

1. Richardson, B.B., *Aggreting Emaıl*. **2017,** (US Patent 20,170,279,756)
2. Cobb S., The Economics Of Spam. http://www.spamhelp. org/articles/economics of spam. pdf (Ocak **2018**)
3. Halawa, H., Ripeanu, M. Beznosov, K., Coskun, B., Liu, M., An Early Warning System for Suspicious Accounts. In Proceedings of AISec'17, Dallas, TX, USA, **2017**
4. Internet Security Threat Report. https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf, (Şubat **2018**)
5. Andress, J., Winterfeld S., *Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners*. 2 ed.: ELSEVIER.
6. GAO, K. What is Email Spam? https://emailmarketing.comm100.com/email-marketing-ebook/email-spam.aspx. (Eylül, **2016)**
7. GAO, K. CAN-SPAM Compliance: What Is CAN-SPAM & Why Does It Matter? https://emailmarketing.comm100.com/email-marketing-tutorial/can-spam-compliance.aspx. (Ekim **2017**)
8. Brownlee, J. Machine learning mastery. http://machinelearningmastery.com/discover-feature-engineering-howtoengineer-features-and-how-to-getgood-at-it (Nisan, **2016**)
9. Bayes Sınıflandırıcılar.http://mail.baskent.edu.tr/~20410964/DM_9.pdf (Mart, **2016)**
10. Miner, R. Naive Bayes. https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/bayesian/naive_bayes.html (Ocak, **2016**)
11. 6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R), https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (Şubat, **2016**)
12. Miner, R. Naive Bayes (Kernel), https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/bayesian/naive_bayes_kernel.html (Ocak, **2016)**
13. Breheny, P. Kernel Density Classification, https://web.as.uky.edu/statistics/users/pbreheny/621/F12/notes/10-25.pdf (Ocak, **2016)**
14. Ayhan, S., Ş. Erdoğmuş, Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi,* **2014**
15. Athanasopoulos, A., et al. GPU acceleration for support vector machines. in Procs. 12th Inter. Workshop on Image Analysis for Multimedia Interactive Services, *WIAMIS 2011, Delft, Netherlands*. **2011.**
16. Chang, C.-C. and C.-J. Lin, LIBSVM: a library for support vector machines, *ACM transactions on intelligent systems and technology (TIST)*, **2011**, p. 27.
17. Shalev-Shwartz, S., et al., Pegasos: Primal estimated sub-gradient solver for svm. Mathematical programming, **2011**, p. 3-30.
18. Albayrak, A.S., YILMAZ Ö, Veri madenciliği: Karar ağacı algoritmaları ve İMKB verileri üzerine bir uygulama. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **2009**.
19. Safavian, S.R. and D. Landgrebe, *A survey of decision tree classifier methodology*. IEEE transactions on systems, man, and cybernetics, **1991**, p. 660-674.

20. Şengöz, N. Yapay Sinir Ağları, http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/ (Ocak, **2017**)
21. Elmas, Ç., *Yapay Zeka Uygulamaları:(Yapay Sinir Ağı, Bulanık Mantık, Genetik Algoritma),* Seçkin Yayıncılık, **2011**
22. Kabalcı, E., Çok Katmanlı Algılayıcı (Multilayer Perceptron). https://ekblc.files.wordpress.com/2013/09/mlp.pdf, (Ocak, **2017**)
23. Taşcı, E. and A. Onan, K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi. *Akademik Bilişim*, **2016**
24. Bulut, F., Amasyalı F., En Yakın k Komşuluk Algoritmasında Örneklere Bağlı Dinamik k Seçimi. *Akıllı Sistemlerde Yenilikler ve Uygulamaları (ASYU2014)*, **2014**
25. Brownlee, J. A Gentle Introduction to the Bag-of-Words Model https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (Ocak, **2016**)
26. Soumya, G., Joseph S., Text classification by augmenting bag of words (bow) representation with co-occurrence feature. I*OSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p-ISSN: 2278-8727 Volume. 16: p. 34-38.
27. Cianflone, A. and L. Kosseim, N-gram and Neural Language Models for Discriminating, *Similar Languages Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects, pages 243–250, Osaka, Japan, **2016**.*
28. Sah, U.K. and N. Parmar, *An approach for Malicious Spam Detection In Email with comparison of different classifiers.* **2017**.
29. Renuka, D.K. and P.V.S. Rajamohana, An Ensembled Classifier for Email Spam Classification in Hadoop Environment. *Appl. Math*, **2017,** p. 1123-1128.
30. Kumagai, A. and T. Iwata, *Learning Latest Classifiers without Additional Labeled Data. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17),* **2017**
31. Ajaz, S., M. Nafis, and V. Sharma, Spam Mail Detection Using Hybrid Secure Hash Based Naive Classifier, *International Journal of Advanced Research in Computer Scienc*e, **2017**.
32. Lee, C.-N., Y.-R. Chen, and W.-G. Tzeng. An online subject-based spam filter using natural language features, in *Dependable and Secure Computing, 2017 IEEE Conference on*. **2017**. IEEE.
33. Carreras, X. and L. Marquez, Boosting trees for anti-spam email filtering, *Proceedings of RANLP-2001*, pp. 58-64, Bulgaria, **2001**
34. Chuan, Z., et al., A LVQ-based neural network anti-spam email approach. *ACM SIGOPS Operating Systems Review*, **2005**. p. 34-39.
35. Sharma, A.K. and S. Sahni, A comparative study of classification algorithms for spam email data analysis. *International Journal on Computer Science and Engineering*, **2011**, p. 1890-1895.
36. Weka 3: Data Mining Software in Java. https://www.cs.waikato.ac.nz/ml/weka/ (Ocak, **2016**)
37. Weka-Out of Memory Hatası, https://weka.wikispaces.com/OutOfMemoryException. (Eylül, **2016**)
38. RapidMiner Studio 7.6 Sürümü, https://rapidminer.com/ (Ekim, **2016**)
39. Joshi, R. Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures.http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. (Eylül, **2017**)
40. Brownlee, J. Classification Accuracy is Not Enough: More Performance Measures You Can Use. https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ (Şubat, **2018)**

EKLER

Ek 1: 11th IEEE International Conference (AICT2017) Bildirisi, Moskova, Rusya

Abstract— **With the advent of the Internet and reduction of the costs in digital communication, spam has become a key problem in several types of media (i.e. email, social media and micro blog). Further, in recent years, email spamming in particular has been subjected to an exponentially growing threat which affects both individuals and business world. Hence, a large number of studies have been proposed in order to combat with spam emails. In this study, instead of subject or body components of emails, pure use of hyperlink texts along with word level n-gram indexing schema is proposed for the first time in order to generate features to be employed in a spam/ham email classifier. Since the length of link texts in e-mails does not exceed sentence level, we have limited the n-gram indexing up to trigram schema. Throughout the study, provided by COMODO Inc, a novel large scale dataset covering 50.000 link texts belonging to spam and ham emails has been used for feature extraction and performance evaluation. In order to generate the required vocabularies; unigrams, bigrams and trigrams models have been generated. Next, including one active learner, three different machine learning methods (Support Vector Machines, SVM-Pegasos and Naive Bayes) have been employed to classify each link. According to the results of the experiments, classification using trigram based bag-of-words representation reaches up to 98,75% accuracy which outperforms unigram and bigram schemas. Apart from having high accuracy, the proposed approach also preserves privacy of the customers since it does not require any kind of analysis on body contents of e-mails.**

***Index Terms*— Spam Email, Machine Learning, Active Learning, N-Grams, Bag of Words**

Introduction

Spam has been one of the leading problems in digital communication over the couple of decades. As reported in [1], the most common form of spam is email spamming which refers to receiving unwanted bulk email messages. Due to the very limited cost and great easiness in reaching millions of Internet users, spam has been excessively used by amateur advertisers and direct marketers [10]. On the other hand, spam mails do not only waste the time of Internet users but also causes several harms such as excessive bandwidth usage and potential virus attacks. According to the International Telecommunication Union, 70% of the emails around the world have been sent for spamming purpose [2].

In order to thwart the problem of spam emails, various approaches and methods have been proposed. Tretyakov [3] and Bhowmick & Hazarika [4] have collected these approaches under two headings: (1) *knowledge engineering (KE)* and (2) *machine learning (ML)*. While the former case comprises the methods of heuristic filters, blacklisting, whitelisting, greylisting, collaborative spam filtering, honey pots and signature schemes, latter one deals with classification of spam emails by employing machine learning methods over various extracted features from either content or header sections of emails. Though, they have satisfactory success in spam mail filtering, knowledge engineering based approaches are subjected to some problems and shortcomings. For instance, clear domains can be blocked because of spammer exploits or accounts of innocent users can be attacked by hackers [4]. Moreover, rules in those approaches should be updated frequently in order to adapt new types of threats. Apart from rule based heuristic techniques, ML based approaches achieve more success in filtering/classification results since they do not require any explicit rulesets. Instead, ML methods capture and "learn" the hidden and discriminative patterns contained by spam/ham mails by utilizing of real world training data. As pointed out by [4], spam filtering is difficult due to its dynamic nature. Thus, the approach of ML has led to create more adaptive and dynamic spam countermeasures in recent decade.

To date, literature of spam mail detection has witnessed the increasing number of machine learning based studies which examine various types of machine learning methods on different types of features. For instance, [5] has employed the method of Naive Bayes whereas [6, 7] have examined the performance of Support Vector Machine on spam email classification. Further, the capabilities of decision trees [8] and conventional neural networks [9] have also been addressed.

In machine learning, it is a well-known fact that, apart from the employed method it also plays a key role that which features have been selected and how they have been preprocessed and represented. If the literature is reviewed, it can be seen that bag of words (BOW) and n-gram feature representations come into prominence since emails are composed of heavily textual data (e.g. words and characters). It is very common in literature to take all the words in the corpus and generate BOW vocabulary for further creation of word frequency based sparse feature vectors that will be used to train a ML model. Besides, character or word level n-gram indexes have also been used frequently. According to Moon et al. [11], n-gram indexes remove the necessity of morpheme analysis and word spacing problems. Moreover, n-gram indexes enable language independency.

In their study, Moon et al. [11] proposed a spam/ham mail classification schema by using n-gram indexing and SVM. They founded out that SVM achieves better results compared to other methods

such as Naïve Bayes and *k* nearest neighborhood classification. Blanco et al. [12] have particularly studied the problem of reducing the number of false positives in anti-spam email filters by use of SVM. They proposed an ensemble of SVMs that combines multiple dissimilarities. Dai et al. [13] proposed a system which extracts text based features from body contents and employs an active machine learning method in order to classify malicious spam emails. On the other hand, use of URL (Uniform Resource Locator) has been also incorporated in phishing/spam mail recognition. For instance, Basnet and Doleck [14] have employed URL based features to classify whether a web page is phish or not. They have encoded URLs into 138 dimensional feature vectors and applied several ML methods covering decision trees, random forest, multilayer perceptron and SVM.

As can be seen, various features and different ML methods have been employed in the spam mail classification field. However, according to our best knowledge in spam mail classification field, pure use of hyperlink texts located in email bodies has not been addressed yet. In this study, in contrast to other existing approaches, we propose the use of hyperlink texts located in body of emails as the feature source. In detail, we have employed n-gram indexing schema by extracting word level unigram, bigram and trigram features from a large scale email link corpus collected from 50.000 hyperlinks. Extracted n-grams have been filtered out by a certain frequency threshold and then 3 ML methods (Naïve Bayes, SVM and Pegasos - an online version of SVM) have been employed to train 3 different classifiers. According to the results, link texts found suitable to be used as a spam/ham mail classifier. In this way, it is also enabled to have a privacy preserving classification scheme since our approach does not require full body content. We believe that, the proposed approach is a suitable and robust solution for the companies which care about email privacy.

The organization of the paper is as follows: Section 2 briefly introduces the employed ML methods. Section 3 presents the properties of used dataset. Section 4 presents the preprocessing stages and Section 5 details the conducted experiments along with results and finally Section 6 concludes the study.

Methods

In this study we have employed three different machine learning methods: (1) Naïve Bayes multinomial (NB), (2) SVM [15] and finally (3) SVM Pegasos [16]. We have first tested the performance of offline methods such as NB and SVM and then moved to Pegasos whether the proposed approach is suitable for an online learning schema. These 3 methods have been briefly explained below. Due to the space limitations, further details about the methods have been left to readers.

Naive Bayes (NB)

Naïve Bayes classifiers have been perhaps the most well-known and well-used statistical based classifiers in the field of spam filtering over the years [4]. According to the [5], the main reason of why it is called "Naïve" is that it transforms multivariate problems to a univariate problems by disregarding the possible dependencies or correlations among inputs. As stated in [4], construction and interpretation of NB classifiers are easy and they are effective even for large scale datasets.

Inspired from Bayes' theorem, Bayes classifiers are simple probabilistic classifiers based on strong independence assumptions and they employ a probabilistic approach for inference. Given the class label *y*, a Naïve Bayesian classifier's task is to approximate the class-conditional probability with assumption of conditional independence among the attributes. The conditional independence assumption is given in Eq. 1: [17].

$$P(\boldsymbol{X}|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y) \quad (1)$$

where each attribute set $\boldsymbol{X} = \{X_1, X_2, \ldots, X_d\}$ comprises d attributes. At this point, instead of calculating class-conditional probability for each combination of $\boldsymbol{X}$, it is sufficient to approximate the conditional probability of each X_i, given Y [17].

For classification of an unknown case, NB classifier calculates the posterior probability of every class Y as presented in Eq. 2 below:

$$P(Y|\boldsymbol{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(\boldsymbol{X})} \quad (2)$$

For further reading, readers can see [3] and [17].

Support Vector Machine (SVM)

Support Vector Machines (SVM) developed by V. Vapnik is one of the most widely used ML method due to having solid theoretical background along with good generalization performance in several problem domains. Moreover, SVM is also capable of handling high dimensional data avoiding the curse of dimensionality [17]. As a non-parametric method, the principle idea of SVM is building a maximal margin hyper plane between the positive and negative samples in order to linearly separate them. Thus, a better generalization is obtained. A formal explanation about finding a linear separation boundary is given as below [4,17]:

Let $X = \{(\boldsymbol{x}_i, y_i)\}$, $\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$ corresponds to the training data which has the collection of attributes where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Then the separating hyperplane of a linear classifier can be defined as $\mathbf{w}.\mathbf{x} + b = 0$ where $\mathbf{w}$ and b constitute the parameters of the model. In this way, a separating hyperplane which divides the training data into corresponding classes is achieved (i.e.

sign($w^T x_i + b$) = y_i for all i). At this stage, the margin m_i between a training example and the separating hyperplane can be formulated as in Eq. 3.

$$m_i = \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|} \quad (3)$$

The key point of SVM training phase is to maximize the margin for each example which yields the minimization of the objective function of $f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$ subject to $y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1$ where i = 1, 2, … *N*. At this point, the standard Lagrange multiplier method can be employed to solve this quadratic and convex optimization problem by use of Eq. 4 presented below:

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i \, (y_i(\mathbf{w}.\mathbf{x}_i + b) - 1) \quad (4)$$

As a sum, SVM can be expressed as a constrained quadratic programming problem.

As is known, spam has a dynamic nature and a suitable countermeasure should be adaptive for new unseen cases. Though they present satisfactory classification performance the main shortcoming of offline algorithms such as SVM and NB is that they require training from scratch for new examples. In order to overcome this problem, Shalev-Schwartz et al. [18] have extended SVM in terms of (1) better accuracy, (2) shorter training duration and (3) online learning capability and named their own implementation as "Pegasos". Details of Pegasos algorithm can be found at [18]. According to [18], Pegasos is well suited for text classification problems which involve training with sparse feature vectors. Therefore, we have also conducted experiments by employing the SVM Pegasos in order to achieve an easy to update spam email classification schema since it serves an online learning capability which prevents full model training from scratch.

Dataset

According to our best knowledge, although there exist various datasets in literature, there exists no publicly available large scale spam/ham email hyperlink dataset. Therefore, the spam/ham email dataset that has been utilized during the study has been supplied by a network security company named COMODO Inc. The used dataset involves 150.000 spam and 141.414 ham hyperlinks extracted from 47382 spam and 8506 ham emails collected in two months duration. Supplied dataset comprise 4 attributes: (a) *message-id*, (b) *hyperlink URL*, (c) *hyperlink text* and (d) *spam/ham flag*. As can be seen, dataset does not involve any kind of non-hyperlink information such as subject of message content. In the next stage, whole dataset has been preprocessed in order to generate robust and meaningful n-grams for further analyzes.

Preprocess

Prior to generation of n-gram vocabulary we have employed two phase preprocessing. In the first one, we have checked all instances in spam/ham dataset for the validity of hyperlink texts and found out some inappropriate entries such as null values or HTML/CSS related tags. Therefore, we have cleared those records from the dataset. In this way, the number of spam and ham hyperlinks have been reduced to 146.288 and 132.439 respectively.

At the second stage of preprocessing we have carried out some sub procedures such as stop word removal (e.g. the, www, and etc.) and uppercasing the words in order to obtain a uniform representation. The procedure that we have followed is given in Table 1 as a pseudo code format.

TABLE I. PSEUDO CODE OF THE PREPROCESSING PROCEDURE

```
Input: D: spam and ham mail dataset(in "message-id, hyperlinkURL, hyperlinkText" format)
Output: PD: Preprocessed dataset
foreach line in D
    select hyperlinkText
    If(line has hyperlinkText)
        add to PD
    else continue;
foreach line in PD
    Uppercase the line
    Divide line to word and add wordlist
    If (word have special characters)
        Remove special characters from word
    If (word length <=2)
        Remove from wordlist
    If (word is a special word) //special word for example; "http, www, and etc.)
        Remove from wordlist
```

Following the preprocessing, we have extracted unigrams, bigrams and trigrams by using a specially created C# application. Since the number and quality of the features have a great impact on machine learning models we have preferred to filter out infrequent n-grams with a certain threshold values such as 30, 40 and 50. Thus, different number of n-gram features have been obtained (See Table 2).

TABLE II. TOTAL NUMBER OF N-GRAMS WITH CERTAIN THRESHOLDS

	Number of n-grams		
Threshold	**30**	**40**	**50**
# of 1-grams	3332	2654	2192
# of 2-grams	4466	3347	2648
# of 3-grams	4474	3267	2540

At this stage, 9 different training datasets belonging to each n-gram and threshold configuration

have been generated by use of BOW mapping on detected n-grams. On the other hand, in order to reduce the required training time and hold some remaining data for further validation processes, size of datasets have been reduced to that each contain 50.000 instances by using random sampling technique.

Experiments and results

As it was stated before, one of the main goals of this study is to understand whether the hyperlink texts in emails play a discriminative role in spam/ham classification. In order to validate this hypothesis we have employed 3 different classification method namely Naïve Bayes, SVM and SVM Pegasos.

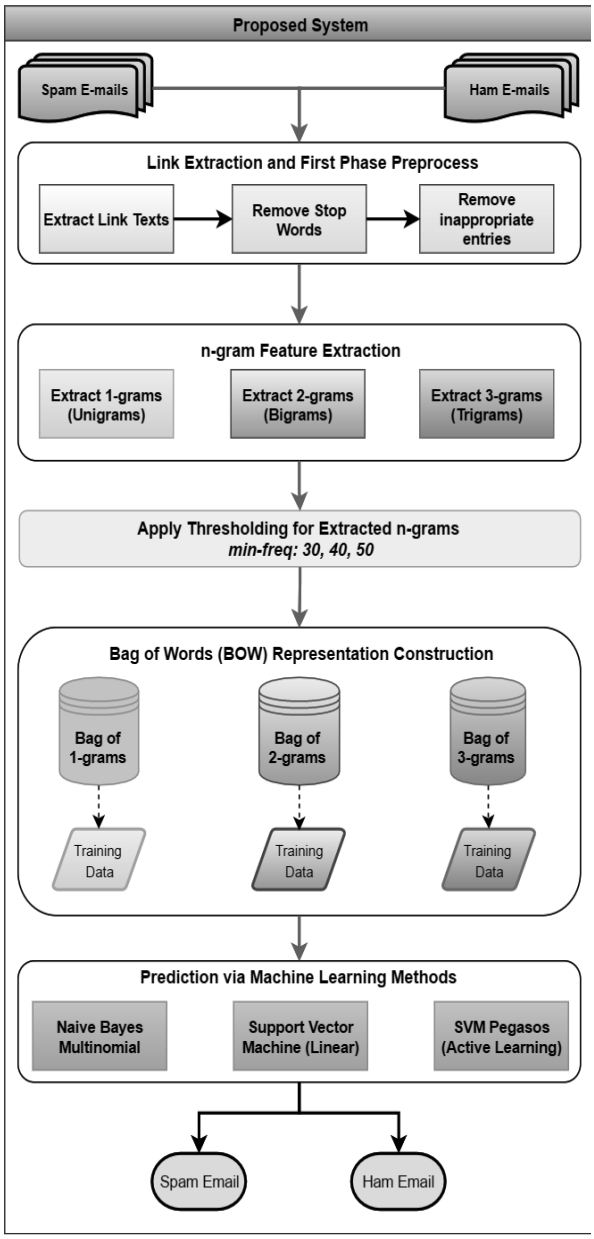


Fig. 1. Workflow of the proposed system

Furthermore, SVM and Pegasos algorithms have been adjusted in order to be employed with linear kernel. Throughout the experiments, we have applied 5 fold validation schema to validate stability and robustness of the created models. The workflow of the study has been depicted in Fig. 1. It should be noted that, training of selected ML models have been carried out in software of Rapid Miner Studio 7.5 because of its easy to use GUI based visual modelling tools. Performance of the models have been presented in Table 3 in terms of accuracy and classification error.

TABLE III. AVERAGE ACCURACY SCORES OF 5 FOLD CROSS VALIDATION

Dataset	**# of n-grams**	**Accuracy of Methods**		
		SVM	Pegasos	NB
30 Threshold Dataset	1 Gram	85.92 %	90.77 %	89.09 %
	2 Gram	93.66 %	96.02 %	95.77 %
	3 Gram	89.48 %	98.60 %	98.45 %
40 Threshold Dataset	1 Gram	86.23 %	90.76 %	88.76 %
	2 Gram	94.64 %	96.01 %	95.74 %
	3 Gram	92.37 %	98.58 %	98.52 %
50 Threshold Dataset	1 Gram	86.50 %	90.43 %	88.58 %
	2 Gram	94.43 %	96.11 %	95.84 %
	3 Gram	90.83 %	**98.75 %**	98.67 %

TABLE IV. AVERAGE CLASSIFICATION ERRORS OF MODELS

Dataset	**# of n-grams**	**Accuracy of Methods**		
		SVM	Pegasos	NB
30 Threshold Dataset	1 Gram	14.08 %	9.23%	10.91 %
	2 Gram	6.34 %	3.98%	4.23 %
	3 Gram	10.52 %	1.40%	1.55 %
40 Threshold Dataset	1 Gram	13.67 %	9.24%	11.24 %
	2 Gram	5.36 %	3.99%	4.26 %
	3 Gram	7.63 %	1.42%	1.48 %
50 Threshold Dataset	1 Gram	13.50 %	9.57%	11.42 %
	2 Gram	5.57 %	3.89%	4.16 %
	3 Gram	9.17 %	1.25%	1.33 %

Results

According to the obtained results, the following findings have been discovered:

- Among the methods that we have employed, SVM Pegasos has achieved the best result (i.e. accuracy of 98.75%) within the trigram configuration on 50 threshold dataset.
- It has been observed that the classification errors tend to decrease with the increment of

threshold value. This finding not only increases the accuracy but also enables building more compact feature vectors.

- Naïve Bayesian classifiers have achieved the least training durations among the employed methods. However, SVM Pegasos seems to be the most feasible method when the capability of online learning is considered.
- Although both SVM and SVM Pegasos have originated from the same computational methodology, it is observed that the improvements in SVM Pegasos enables to collect better results in sparse and linearly separable datasets. Therefore, we can conclude that SVM Pegasos better suits to our problem domain than conventional SVM.
- Apart from the reported findings, we have also repeated the experiments with 10-fold cross validation. Nonetheless, it has been observed that 10 fold cross validation results do not show a significant difference in terms of accuracy.

Conclusıon

In this study, it was aimed to use hyperlink texts found in emails in order to classify spam/ham emails. For this purpose 3 well-used and well-known predictive machine learning methods have been employed. Throughout the study, a novel large scale data set has been utilized. Besides, n-gram indexing schema along with bag of n-grams have been used as the feature generation technique. In order to refine and reduce the large number of features, certain threshold values have been applied.

According to the promising results, hyperlink texts have been found suitable for spam/ham email classification problem. With this achievement, it is made possible for network security companies to develop spam email detection systems without the necessity of full email body and header content analysis. As a future work, it is being planned to use more effective machine learning methods such as Twin SVM or Deep Random Forests.

Acknowledgment

This work is supported by Comodo Group, Inc. and authors would like to thank for their collaboration on dataset collection stage.

References

[1] D. Sculley and G. M. Wachman. “Relaxed Online SVMs for Spam Filtering”. In SIGIR'07, 2007.

[2] S. Teli and S. Biradar. “Effective Spam Detection for Email”, In International Conference on Advances in Engineering & Technology, 2014.

[3] K. Tretyakov. Machine learning techniques in spam filtering. Technical Report, Institute of Computer Science, University of Tartu, 2004.

[4] A. Bhowmick and S.M. Hazarik. Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends, arXiv:1606.01042, 2016.

[5] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz. “A Bayesian Approach to Filtering Junk Email”. In 15th National Conference on Artificial Intelligence, USA, 1998.

[6] M. Woitaszek and M. Shaaban, “Identifying Junk Electronic Mail in Microsoft Outlook with Support Vector Machine”. In Proceedings of the 2003 symposium on applications and the internet, 2003.

[7] O. Amayri and N. Bouguila. A Study of Spam Filtering using Support Vector Machines, Artificial Intelligence Review 34(1):73-78, 2010.

[8] F. Toolan and J. Carthy. “Feature Selection for Spam and Phishing Detection”, In eCrime Researchers Summit, 2010.

[9] C. Wu, Behaviour-based Spam Detection using a Hybrid Method of Rule-based Techniques and Neural Networks. Expert System with Applications 36(3): 4321-4330, 2009.

[10] L.F. Cranor and B.A. Lamacchia, Spam!, Communications of the ACM 41(8), 1998.

[11] J. Moon, T. Shon, J. Seo, “An Approach for Spam E-Mail Detection with Support Vector Machine and N Gram Indexing”, In International Symposium on Computer and Information Sciences, ISCIS pp 351-362, 2004.

[12] A. Blanco, A. M. Ricket and M. M. Merino, “Combining SVM Classifiers for Email Anti-spam Filtering”, International Work-Conference on Artificial Neural Networks IWANN, Computational and Ambient Intelligence pp 903-910, 2007.

[13] Y. Dai, S. Tada, T. Ban, J. Nakazato, J. Shimamura and S. Ozawa, “Detecting Malicious Spam Mails: An Online Machine Learning Approach”, International Conference on Neural Information Processing ICONIP Neural Information Processing pp 365-372, 2014.

[14] R.B. Basnet and T. Doleck, “Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach”, In: 2015 International Conference on Computational Intelligence & Communication Technology, 2015.

[15] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[16] S. Shwartz, Y. Singer, N. Srebro “Pegasos: Primal Estimated sub-Gradient Solver for SVM” ,24th International Conference on Machine Learning, Corvallis, 2007

[17] P. N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison Wesley, USA, 2005.

[18] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, “Pegasos: Primal Estimated sub-Gradient Solver for SVM” ,24 th International Conference on Machine Learning, Corvallis, 2007

***Özetçe*—Günümüzde elektronik ortamda sıkça kullandığımız iletişim yollarından birisi olan e-postalar; kişisel iletişimler, iş odaklı aktiviteler, pazarlama, reklam, eğitim vs. gibi birçok nedenden dolayı hayatımızda önemli bir yer kaplamaktadır. Birçok farklı konudaki iletişim ihtiyacı için hayatımızı kolaylaştıran e-postalar, kötü amaçla kullanıldıklarında alıcıların zamanını çalması, maddi ve manevi kayıplara sebep olması nedeniyle hayatı oldukça zorlaştırabilmektedir. Tanımadığımız ya da güvenilmeyen adreslerden, reklam ya da güvenlik tehdidi oluşturma amaçlı gönderilen e-postalar, bilgi güvenliği açısından önemli tehlikeler oluşturabilir. Bu türdeki istenmeyen e-postaları, insanlara zarar vermeden önce tespit edip önleyebilmek ise ayrı bir çalışma konusu olmuştur.**

Bu çalışmada, literatürdeki çalışmalardan farklı olarak e-posta içeriğinde yer alan linklerin metinleri ele alınarak, makine öğrenmesi yöntemleri ve Kelime Kümesi Tekniği ile istenmeyen e-posta/e-posta sınıflaması yapılmıştır. Yapılan çalışmada "doğruluk" metriği kullanılarak Kelime Kümesi Tekniği sonucu oluşturulan farklı N-Gramların sınıflandırma başarısına olan etkisi ve farklı makine öğrenme tekniklerinin istenmeyen e-posta sınıflandırılmasındaki başarısı analiz edilmiştir.

Anahtar Kelimeler —Spam Filtreleme, Kelime Kümesi Tekniği, İstenmeyen E-posta, Makine Öğrenmesi

***Abstract*—Nowadays, we use frequently e-mails, which is one of the communication channels, in electronic environment. It play an important role in our lives because of many reasons such as personal communications, business-focused activities, marketing, advertising, education, etc. E-mails make life easier because of meeting many different types of communication needs. On the other hand they can make life difficult when they are used outside of their purposes. Spam emails can be not only annoying receivers, but also dangerous for receiver's information security. Detecting and preventing spam e-mails has been a separate issue.**

In this study, Unlike the studies in the literature; the texts of the links which is in the e-mail body are handled and classified by the machine learning methods and the Bag of Word Technique. we analyzed the effect of different N-Grams on classification performance and the success of different machine learning techniques in classifying spam e-mail by using "accuracy" metric.

Keywords — Spam Filtering, Bag of Word Technique, Spam Email, Machine Learning

Giriş

Elektronik ortamda birçok farklı konudaki iletişim ihtiyacı için hayatımızı kolaylaştıran e-postalar, asıl amacı kullanıldıklarında alıcıların zamanını çalması, maddi ve manevi kayıplara sebep olması nedeniyle hayatı oldukça zorlaştırabilmektedir. Tanımadığımız ya da güvenilmeyen adreslerden, reklam ya da güvenlik tehdidi oluşturma amaçlı gönderilen e-postalara, "istenmeyen e-posta" (*Spam*) denilmektedir [1]. Özellikle pazarlama ve reklam yöntemi olarak çok sayıda işletme potansiyel müşterilerinin mail adreslerini toplamayı denemektedir. Bu işletmelerin bazıları kötü niyetli davranarak topladığı bu bilgileri diğer ticari işletmelerle paylaşmaktadır [1]. Bunun sonucunda binlerce insan istenmeyen içerikli maillerle karşı karşıya kalmaktadır. Stephen tarafından yapılan durum çalışmasında istenmeyen e-posta ekonomisinin getirdiği kar, gerçek hayattaki şu örnekle anlatılmıştır [2]: Geleneksel reklam yaklaşımı ile potansiyel bir müşteriye gönderdiğiniz her broşür, en az 1.00 USD değerindedir. 5.000 broşürü postaladığımızı düşünürsek sadece posta maliyeti 5.000 dolar etmektedir. Toplu e-posta yolu ile reklam yapmak ise durumu oldukça farklı bir hale getirir. Wall Street Journal'ın (11/13/02) yaptığı bir araştırma, e-postaları kullanırken % 0.001 gibi düşük bir getiri oranının karlı olabileceğini göstermektedir. 5 milyon mesajın gönderildiği bu örnekte, ilk hafta %0.0023 gibi bir oranla 81 satışla sonuçlanmış, toplamda 1.500 doları gelir elde edilmiştir. Buradaki 5000 iletiyi gönderme maliyeti çok düşüktür. 56 Kbps modem üzerinde bile saatte binlerce mesaj gönderebilir. Bu durum çalışmasında da görüldüğü gibi istenmeyen e-postalar, düşük maliyetle reklam ve pazarlama ile yüksek kazançlar sağlamaktadır.

İstenmeyen e-postalar bilgi güvenliği açısından da önemli tehlikeler oluşturabilir. Bu tehlikeyi insanlara zarar vermeden önce tespit edip önleyebilmek ise ayrı bir çalışma konusu olmuştur. 2017 yılı İnternet güvenliği tehdit raporuna bakıldığında e-postaların yarısından fazlasının (%53) istenmeyen e-posta, bu e-postaların artan bir kısmının da kötücül amaçlı yazılımlar *(malware)* içerdiği görülmektedir [3]. 2015 yılında 220 e-postadan 1 tanesinin kötücül amaçlı yazılım içerdiği görülürken, 2016 yılında bu oran 131 e posta da 1'e yükselmiştir [3]. E-posta ile bulaşan kötücül amaçlı yazılımlardaki bu artış, bu tür yazılımların büyük oranda profesyonelleştiğini göstermektedir. Bir diğer yandan ise saldırganların bu tür saldırılardan önemli kazançlar sağladıklarını

ve bu tarz istenmeyen e-postaların etkilerinin devam etmesinin muhtemel olduğunu göstermektedir [3].

İstenmeyen e-postaların burada bahsedilen tehlikelerinden korunmak için iyi bir filtreleme ihtiyacı doğmuştur. İstenmeyen e-postaları sınıflandırmak için yapılan çalışmalar incelendiğinde farklı farklı teknikler ile e-postaların başlık(header), içerik (body) bölümlerindeki çeşitli bilgiler kullanılarak sınıflandırma yapıldığı görülmüştür. Örneğin Sah ve arkadaşları [4] tarafından yapılan çalışmada sırasıyla veri toplama işlemi, verileri ön işleme, verilerden özellik çıkarma, verileri sınıflandırma ve sonuçları analiz etme işlemi yapılarak metin(body) tabanlı sınıflandırma çalışması yapılmıştır. Çalışmada eğitim için 702 e-posta, test için 260 e-posta kullanılarak Destek Vektör Makineleri *(Support Vector Machine (SVM))* ve Naive Bayes algoritmaları ile sınıflandırma yapılmıştır. Çalışma sonucunda Naive Bayes ve SVM algoritmalarının benzer başarı gösterdiği sonucuna varılmıştır. Benzer bir diğer çalışma Renuka ve arkadaşları [5] tarafından Hadoop ortamında yapılan sınıflandırma çalışmasında istenmeyen e-postaları ayırt etmek için Gradient Boost ve Naive Bayes sınıflandırma tekniği kullanılmıştır. Çalışma temelde eğitim ve test olmak üzere iki aşamada ele alınmıştır. Çalışma da sınıflandırma performansını iyileştirmek için tek düğüm Hadoop ortamı kullanılarak farklı sınıflandırma tekniklerinin bir arada kullanıldığı karma bir model çıkarılmıştır. Duyarlılık*(precision)*, doğruluk(*accuracy*) ve hassasiyet*(recall)* metrikleri ile çalışmanın performansı ölçülmüştür.

Literatürdeki çalışmalar ve bu çalışmalarda yer alan bilgilere göre istenmeyen e-posta/ e-posta sınıflandırma çalışmaları ile ilgili genel olarak şu sonuçlar çıkartılmaktadır:

- Kötü amaçlı istenmeyen e-posta araştırmalarının çoğunda araştırmacılar, istenmeyen e-postaları ayırt etmek için makine öğrenme algoritmalarının eğitiminde kullanılacak olan içerik temelli yaklaşımlara yönelmişlerdir. Bu yaklaşımlar 4 ana kategoriye ayrılabilir. Bunlar; başlık *(header)* özellikleri, konu *(subject)* özellikleri, e-posta metni*(body)* özellikleri ve e-posta eki*(attachment)* özellikleridir [7].
- Çalışmalarda çok kullanılan makine öğrenmesi teknikleri; SVM, ANN, RF, Naive Bayes ve AdaBoost olmuştur [7]. Bazı araştırmalarda ise bilinen makine öğrenme tekniklerinin yanı sıra Rough setler gibi kural bazlı yaklaşımları da konu alan karma yaklaşımlar sergilenmiştir [7].
- Çalışmalarda genel olarak kullanılan e-posta verileri Spambase(1999), Spam Assassin(2006), TREC(2007)'dir [7].

Literatürdeki çalışmalardan farklı olarak bu çalışmanın temel motivasyonu; e-postaların içeriklerinde yer alan bağlantıların metinlerine odaklanmasıdır. Çalışmada veri setlerinin oluşturulması aşamasında Kelime Kümesi Tekniği (BOW) [22] , sınıflandırma işlemleri için ise makine öğrenmesi teknikleri kullanılmıştır. BOW sonucu oluşan farklı uzunluktaki N Gram'ların sınıflandırma performansına etkisi ve farklı makine öğrenme tekniklerinin istenmeyen e-posta sınıflandırması için başarısı analiz edilmiştir. Çalışma sonucunda Bayes, SVM algoritmalarının en iyi performansı sergilediği görülmüştür.

Methodlar

Çalışma kapsamında ele alınan sınıflandırma algoritmaları Tablo 1'deki gibidir; Sayfa sayısındaki sınırlama nedeniyle bu algoritmaların ayrıntıları okuyuculara bırakılmıştır.

TABLO I. ÇALIŞMADA KULLANILAN MAKİNE ÖĞRENME YÖNTEMLERİ

Karar Ağaçları	Decision Tree [8,9]
	Gradient Booested Tree[8,10]
	Decision Stump [8,11]
	Random Tree [8,12]
	Random Forest [8,13]
Bayes	Naive Bayes[14,15]
	Naive Bayes Kernel [14,15]
Sinir Ağları	Perceptron [16]
Destek Vektör Makineleri	Lib SVM [17]
	Linear SVM [18]
	S-Pegasos [19]
En Yakın Komşu	K-NN [20,21]

Veriseti

İstenmeyen e-postalar ve e-postalarda yer alan bağlantı(link) bilgileri ve bu bağlantılarda yer alan metin bilgileri benzersiz bir e-posta numarası ile birlikte Tablo-2'deki formatta sunulmuştur.

TABLO II. İŞLENMEMİŞ VERİ FORMATI

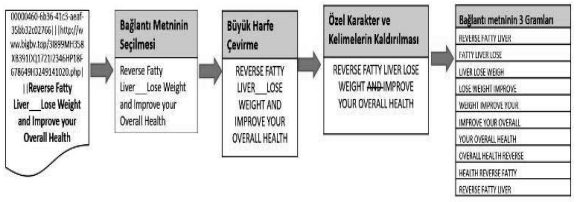
FORMAT
Benzersiz E-Posta Numarası ||| E-Posta Bağlantısı(URL) ||| *Bağlantı Metni*
Örnek: **00000460-6b36-41c3-aeaf 35bb32c02766**|||http://www.bigbv.top/5D899TU358EM391XL1721W2346IN18T678649B3249141020.php|||*Click Here*

İşlenmemiş veriler üzerinde şu işlemler yapılmıştır:

- Her bir satır verinin benzersiz e-posta numarası, bağlantı ve bağlantı metni olarak parçalanmıştır.
- Bağlantı metinlerinde yer alan rakamların ve tanımlanamayan karakterlerin silinmiştir.
- İki karakter olan kelimeler performansı artırmak amacıyla silinmiştir.

- Bazı özel anlam içeren kelimelerin ayırt edici niteliği olmadığı için çıkarılmıştır. Bunlar, Türkçe ve İngilizce haftanın günleri, günlerin kısaltması, Türkçe ve İngilizce yılın ayları, ayların kısaltması, para birimi kısaltmaları, "WWW, HTTP, HTTPS, COM, AND, WITH, THE, İLE, VIA, ANY, YOU, THEY, ARE vs." gibi kelimelerdir. Bu kelimelere veri seti incelenerek karar verilmiştir.
- Türkçe ve İngilizce alfabelerinin farklılığından dolayı oluşan hataları en aza indirmek için tüm harfler büyük harfe çevrilmiştir. Türkçe de yer alan noktalı harfler, noktasız hale dönüştürülmüştür. Bu durum aynı kelime olup birinde noktalı, diğerinde noktasız yazılmış karakterler nedeniyle kelimelerin farklı algılanmasının önüne geçmektedir.

Ön işlemden geçirilmiş olan verilerin, makine öğrenmesi yöntemlerinde kullanılması için eğitim ve test işlemlerinde kullanılacak matris yapıdaki veri seti haline getirilmesi gerekmektedir. Bu amaçla Kelime Kümesi Tekniği (Bag of Word (BOW)) [22] kullanılarak verilerin 1 Gram, 2 Gram, 3 Gram, 4 Gram ve 5 Gram olmak üzere özellikleri(*feature*) çıkarılmıştır. Oluşturulan bu gramlar veri setindeki sütunları oluşturmaktadır. İstenmeyen e-postalar ve e-postalar için ayrı ayrı özellik çıkarım işlemi yapılmıştır. Ayrı ayrı oluşturulan bu özellik kümeleri birleştirilerek ortak olan Gram'lardan tek bir tanesi özellik olarak sayılmıştır. Oluşturulan her özellik, diğer bir deyişle her bir gram özellik vektörünün bir elemanını oluşturmaktadır. Şekil 1'de veri ön işleme ve gramların oluşturulması bir örnekle gösterilmiştir.



Şekil 1. Veri Ön İşleme ve Gramların Oluşturulması

Gramlar oluşturulduktan sonra eğitim ve test işlemlerinin performansını artırmak amacıyla özellik vektörünün boyutu 30, 40 ve 50 limiti belirlenerek azaltılmıştır. Buradaki limitler bir Gram'ın ham veri seti içerisindeki tekrar sayısını göstermektedir. Tekrar sayısı 30'un altında olan gramlar özellik vektörüne dahil edilmemiştir. Özellik kümesinin boyutunun bu şekilde farklı limitlere göre azaltılması aynı zamanda özellik sayısının makine öğrenme teknikleri başarısına olan etkisini gösterecektir.

Özellik vektörü oluşturulduktan sonra veri setinin oluşturulması aşaması Tablo 3'deki örnekteki gibidir. Burada bağlantı metninin her bir gramın özellik vektörü ile karşılaştırılmakta ve tekrar sayısı sayılmaktadır.

TABLO III. VERİ SETİNİN OLUŞTURULMASI

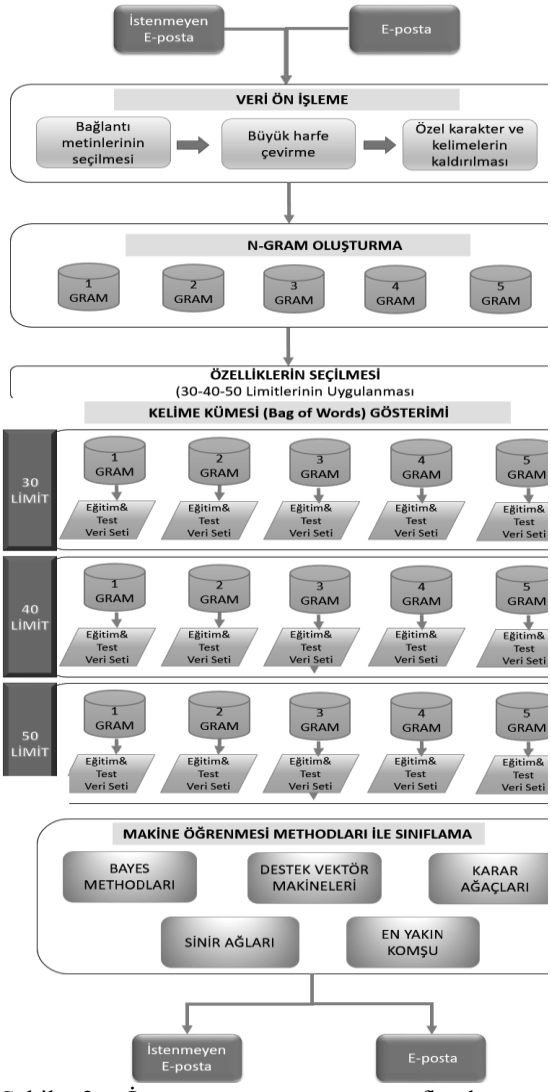
Bağlantı Metni	**Etiket**	**YOUR OVERALL HEALTH**	**FATTY LIVER REMEDY**	**LIVER REMEDY CLICK**	**REMEDY CLICK HERE**	**REVERSE FATTY LIVER**
REVERSE FATTY LIVER LOSE WEIGHT AND IMPROVE YOUR OVERALL HEALTH	İst. EP.	1	0	0	0	2

Deneyler ve Sonuçları

Şekil 2'de istenmeyen e-posta tespiti için önerilen prosedüre ait iş akışı sunulmuştur. Burada veri ön işleme aşamasının ardından N-Gramlar oluşturulmuş, özellik seçimi yapılmış ve oluşan özellik vektörü sınıflandırma algoritmalarına girdi olarak sunulmuştur. Toplamda 15 adet veri seti elde edilmiştir. Veri setlerinin boyutları Tablo 4'deki gibidir. Veri setlerinin boyutları [Bağlantı metni sayısı]X[Özellik Vektörü Boyutu] formatında verilmiştir. Veri setleri oluşturulduktan sonra RapidMiner Studio[8] aracı ile Tablo 1'de belirtilen makine öğrenmesi yöntemleri ile deneyler yapılmıştır. Tüm deneyler 10 katlamalı çapraz doğrulama *(cross validation 10 folds)* ile gerçekleştirilmiştir.

TABLO IV. VERİ SETLERİNİN BOYUTLARI

	1 Gram	**2 Gram**	**3 Gram**	**4 Gram**	**5Gram**
30 LİMİTİ	50000 X 3335	50000 X 4332	44864 X 4286	33818 X 3902	28802 X 3487
40 LİMİTİ	50000 X 2658	50000 X 3239	43045 X 3131	32613 X 2825	25673 X 2525
50 LİMİTİ	50000 X 2188	50000 X 2550	17324 X 2448	32368 X 2192	23232 X 1941



Şekil 2. İstenmeyen e-posta sınıflandırma yöntemi

Deneyler yapılırken öncelikle 50 limiti ile sınırlandırılmış 3 gramlık veri seti ile eğitim yapılmıştır. Bu veri setinin seçilmesinin nedeni özellik sayının diğer veri setlerine göre daha az olması ve gram olarak ortalama bir değerde olmasıdır. Bu eğitimin sonunda başarısı %90'ının üzerinde olan algoritmalar için N gramlara göre performans değişimi incelenmiştir. %90 başarının altında kalan algoritmalarla ise sadece tek bir deney gerçekleştirilmiştir.

Deney sonuçları

Her bir makine öğrenmesi algoritması için doğruluk metriği ile performans ölçümü yapılmıştır. Tablo 5'deki deney sonuçlarına göre şu sonuçlar elde edilmiştir;

- Tablo 5'de belirtilen algoritmalar içerisinde Naive Bayes Kernel, Linear SVM ve Pegasos SVM %99.8 başarı oranı ile en istenmeyen e-postaların sınıflandırılmasında en başarılı algoritmalar olmuşlardır. İstenmeyen e-postaların sınıflandırılması için genel bir bakış açısıyla algoritmalara bakıldığından Bayes Algoritmaları, SVM'ler, K-NN ve Perceptron algoritmalarının oldukça iyi performans gösterdiği görülmektedir. Diğer yandan bir diğer sınıflandırma algoritması olan Karar Ağaçları'nın bu konuda yeterince iyi olmadığı görülmüştür. Bu nedenle Karar Ağaçları için deneyler sadece 50 limit 3 Gram veri seti ile yapılmıştır. N gramların performansa etkisi incelenmemiştir.
- Deney sonuçları genel olarak değerlendirildiğinde 30 limit ile hazırlanan veri setlerinin performansının oldukça düşük olduğu görülmektedir. Bu durum özellik vektörü boyutunun fazla uzun olmasının performansı kötü etkilediğini göstermektedir.
- Gram sayılarının performansa etkisi incelendiğinde neredeyse tüm algoritmalarda gram sayısının artırılmasının doğruluk oranına iyi yönde katkı yaptığı görülmüştür.
- Naive Bayesian sınıflandırıcıları, tüm yöntemler arasında en az eğitim süresine ulaşmıştır. Diğer yandan karar ağaçları ve en yakın komşu algoritması oldukça uzun sürelerde sonuçlanmıştır.

TABLO V. DENEY SONUÇLARI

Veri seti		**Methodların Doğrulukları**						
	N Gram	**NB**	**NBK**	**LibS.**	**LinS.**	**Peg.**	**Prcp**	**K-NN**
Limit=30	1 Gram	78.03	89.31	85.89	91.04	90.80	86.54	90.20
	2 Gram	86.16	95.48	94.45	95.91	95.67	92.95	93.64
	3 Gram	95.84	98.60	89.24	98.60	98.66	95.60	96.75
	4 Gram	97.91	99.38	81.16	99.41	99.38	98.34	98.70
	5 Gram	98.80	**99.89**	99.76	99.89	99.88	99.63	99.72
Limit=40	1 Gram	77.97	89.19	85.97	90.97	90.55	85.38	90.27
	2 Gram	85.69	95.50	94.47	95.88	95.76	93.18	93.62
	3 Gram	95.67	98.73	90.17	98.20	98.76	96.83	96.79
	4 Gram	97.71	99.39	92.00	99.38	99.37	98.01	98.71

	5 Gram	98.67	**99.89**	90.89	**99.89**	**99.89**	99.70	99.70
Limit=50	1 Gram	78.37	89.16	86.62	90.78	90.29	86.40	90.16
	2 Gram	85.54	95.59	94.63	95.93	95.85	93.26	93.22
	3 Gram	95.67	98.79	91.24	98.81	98.80	97.59	96.74
	4 Gram	97.66	99.37	91.81	99.37	99.36	97.94	98.73
	5 Gram	98.53	99.88	91.94	99.88	99.88	99.66	99.66

Karar Ağaçları	Gra. Boosted Trees	Decision Stump	Random Tree	Random Forest
86.49	79.19	63.26	59.10	59.41

Sonuç

Bu çalışma kapsamında farklı N-Gramların performansa etkisi, özellik vektörü boyutunun performansa etkisi ve farklı makine öğrenmesi methodlarının istenmeyen epostaları sınıflamadaki performansı incelenmiştir. Çalışma öncelikle 1,2,3 Gramlık Özellik Vektörleri ve Support Vector Machines, SVM-Pegasos ve Naive Bayes [23] algoritmaları ile yapılmış, daha sonra ise farklı makine öğrenme tekniklerinin karşılaştırılması ve 4 ve 5 Gram ile hazırlanan özellik vektörlerinin performansa etkisinin incelenmesi amacı ile geniş kapsamda ele alınmıştır.

Kaynaklar

[1] Richardson, B.B., *AGGREGATING EMAIL*. 2017, US Patent 20,170,279,756

[2] Cobb, S., *The economics of spam.* ePrivacy Group, http://www. spamhelp. org/articles/economics of spam. pdf, 2003.

[3] *Internet Security Threat Report*. 2017. p. 77.

[4] Sah, U.K. and N. Parmar, *An approach for Malicious Spam Detection In Email with comparison of different classifiers.* 2017.

[5] Renuka, D.K. and P.V.S. Rajamohana, *An Ensembled Classifier for Email Spam Classification in Hadoop Environment.* Appl. Math, 2017. **11**(4): p. 1123-1128.

[6] Sah, U.K. and N. Parmar, *An approach for Malicious Spam Detection In Email with comparison of different classifiers.* 2017.

[7] *RapidMiner Studio 7.6 Sürümü*. 2017; Available from: https://rapidminer.com/.

[8] Safavian, S.R. and D. Landgrebe, *A survey of decision tree classifier methodology*. IEEE transactions on systems, man, and cybernetics, 1991. **21**(3): p. 660-674.

[9] Miner, R. *Decision Tree*. 2017 [12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/parallel_decision_tree.html*

[10] Miner, R. *Gradient Boosted Tree*. 2017 12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient_boosted_trees.html*

[11] Miner, R. *Decision Stump*. 2017 12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/decision_stump.html*

[12] Miner, R. *Random Tree*. 2017 12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/random_tree.html*

[13] Miner, R. *Random Forest*. 2017 12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/parallel_random_forest.html*

[14] Breheny, P. *Kernel Density Classification*. Available from: https://web.as.uky.edu/statistics/users/pbreheny/621/F12/notes/10-25.pdf.

[15] Miner, R. *Naive Bayes Kernel*. 2017 12.12.2017]; Available from: *https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/bayesian/naive_bayes_kernel.html*

[16] Elmas, Ç., *Yapay zeka uygulamaları:(yapay sinir ağı, bulanık mantık, genetik algoritma)*. 2011: Seçkin Yayıncılık.

[17] Chang, C.-C. and C.-J. Lin, *LIBSVM: a library for support vector machines.* ACM transactions on intelligent systems and technology (TIST), 2011. **2**(3): p. 27.

[18] AYHAN, S. and Ş. ERDOĞMUŞ, *Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi.* Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi, 2014. **9**(1)

[19] Shalev-Shwartz, S., et al., *Pegasos: Primal estimated sub-gradient solver for svm.* Mathematical programming, 2011. **127**(1): p. 3-30.

[20] Taşcı, E. and A. Onan, *K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi.* Akademik Bilişim, 2016.

[21] BULUT, F. and M.F. AMASYALI, *En Yakın k Komşuluk Algoritmasında Örneklere Bağlı Dinamik k Seçimi.*
[22] Brownlie J. "*A Gentle Introduction to the Bag-of-Words Model" Kasım 2017,* "https://machinelearningmastery.com/gentle-introduction-bag-words-model/
[23] Bozkir, A.S., Şahin E.., *Spam E-Mail Classification by Utilizing N-Gram Features of Hyperlink Texts.*

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı: Esra ŞAHİN

Doğum Yeri: Kayseri

Medeni Hali: Bekâr

E-posta: esrasahince@gmail.com

Adresi: Mehmet Akif Ersoy Mah. 296. Cad. No: 16 Yenimahalle-Ankara

Eğitim

Lise: Kocasinan Sami Yangın Anadolu Lisesi

Lisans: Gazi Üniversitesi, Bilgisayar Mühendisliği

Gazi Üniversitesi, Elektrik-Elektronik Mühendisliği (Çift Anadal)

Y. Lisans: Hacettepe Üniversitesi, Bilgisayar Mühendisliği

Doktora: -

Yabancı Dil ve Düzeyi

İngilizce, İyi seviyede

İş Deneyimi

Aselsan A.Ş. Kalite Mühendisi	2012-2016
Aselsan A.Ş. Yazılım Mühendisi	2016-Devam

Deneyim Alanları

Yazılım Kalite Yönetimi, Makine Öğrenmesi, Bilgi Güvenliği

Tezden Üretilmiş Projeler ve Bütçesi

Yoktur.

Tezden Üretilmiş Yayınlar

1. Bozkir, A.S., Sahin, E., Aydos, M., Akcapinar Sezer, E., Fatih, O., Spam E-Mail Classification by Utilizing N-Gram Features of Hyperlink Texts, The 11th IEEE International Conference AICT2017, Moscow, Russia 2017
2. Sahin E., Aydos M., "Makine Öğrenme Yöntemleri Ve Kelime Kümesi Tekniği İle İstenmeyen E-Posta / E-Posta Sınıflaması", 26th IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (IEEE CIT-2017) İzmir, Türkiye, 2-5 Mayıs, 2018 (Bildiri değerlendirilme aşamasında).

Diğer Yayınlar

3. Şahin, E., et al. Traffic planning application made by using artificial intelligence (TPAUAI). in Signal Processing and Communications Applications Conference (SIU), 2012 20th. 2012. IEEE.
4. Sahin, E., I.K. Kaynak, and M.U. Sencan. A pilot study: Opportunities for improving software quality via application of CMMI measurement and analysis. in Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on. 2013. IEEE.
5. Sahin, E., I. Keskin, and H. Koç. CMMI-DEV Seviye-3 Sertifikasyonuna Sahip Bir Organizasyonda SCRUM Çevik Yazılım Geliştirme Yöntemi'nin Yazılım Geliştirme Çalışmalarında Uygulanması. in UYMS. 2013.
6. Yalçıner, B.Sonmez S.,Şahin E. et al., "Yazılım güvenlik testi: bir sistematik literatür haritalaması", Ulusal Yazılım Mühendisliği Sempozyumu, 2016

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 09.03.2018

Tez Başlığı / Konusu: MAKİNE ÖĞRENME YÖNTEMLERİ VE KELİME KÜMESİ TEKNİĞİ İLE İSTENMEYEN E-POSTA / E-POSTA SINIFLAMASI

Yukarıda başlığı gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 68 sayfalık kısmına ilişkin, 09/03/2018 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 5'dir.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç/dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

Adı Soyadı: ESRA ŞAHİN — 13.03.2018

Öğrenci No: N14125265

Anabilim Dalı: BİLGİSAYAR MÜHENDİSLİĞİ

Programı: BİLGİSAYAR MÜHENDİSLİĞİ

Statüsü: ☒ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Yrd. Doç. Dr. Murat AYDOS

(Unvan, Ad Soyad, İmza)