Derin Öğrenme Modelleri ile Kimlik Avı E-posta Tespiti

Phishing E-mail Detection with Deep Learning Models

Şeydanur AHİ
Gebze Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
seydanurahi@gtu.edu.tr
ORCID: 0000-0001-8511-440X

İbrahim SOĞUKPINAR
Gebze Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
ispinar@gtu.edu.tr
ORCID: 0000-0002-0408-0277

Öz

Sosyal mühendislik, teknolojiyi kullanarak ya da teknolojiyi kullanmadan insanlardan bilgi edinme (aldatma) sanatıdır. Günümüzde karşı karşıya olduğumuz saldırıların çok büyük bir kısmı insan kaynaklıdır ve aynı şekilde sistemleri değil onları kullanan insanları hedef almaktadır. Güvenlik zincirindeki en zayıf halka olan insan, farklı zamanlarda farklı davranışlar sergilemesinden dolayı güvenlik sürecinde çeşitli zafiyetler gösterebilmektedir. Kimlik avı teknik olarak tüketicilerin finansal veya kişisel bilgilerini ele geçirmek için oluşturulmuş bir tür sosyal mühendislik saldırısıdır. Kimlik avı bugün e-ticaret dünyasının karşılaştığı en büyük zorluklardan biridir. Kimlik avı saldırıları yüzünden birçok şirket ve birey milyarlarca dolar kaybetmektedir. Kimlik avı saldırılarının bu küresel etkisi artmaya devam edecektir ve bu nedenle tehditleri azaltmak için daha etkili kimlik avı algılama tekniklerinin geliştirilmesi gerekmektedir. Bu çalışmada, kimlik avı e-posta saldırılarına karşı derin öğrenme modelleri kullanılarak oluşturulan bir tespit yöntemi önerilmiştir. Önerilen yöntemde gelen e-posta iletilerinin başlık ve gövde bölümlerinden elde edilen özellikler kullanılarak çeşitli derin öğrenme modelleri eğitilmiştir. Yapılan testler sonucunda kimlik avı saldırılarına karşı önerilen bu tespit yöntemi %96,84'lük bir başarı oranı elde edilmiştir.

Gönderme ve kabul tarihi: 08.04.2020 - 24.05.2020
Makale türü: Araştırma

Anahtar sözcükler: *Sosyal Mühendislik Saldırısı, Oltalama E-posta Tespiti, Derin öğrenme, Çok Katmanlı Algılayıcı, LSTM, Word2Vec.*

Abstract

Social engineering is the art of getting information (deception) from people with using technology or without using technology. The vast majority of the attacks facing today are human origin, and likewise, these attacks target computer users. Human being who is the weakest link in the security chain shows various weaknesses in the security process, due to human being's variable behavior in different times. Phishing that is a kind of social engineering attack is technically created to capture consumers' financial or personal information. Phishing is the one of the biggest challenges for the e commerce world. Many companies and individuals lose billions of dollars because of phishing attacks. This global impact of phishing attacks will continue to increase therefore, more effective phishing detection techniques need to be developed to reduce threats. A detection method which is created by using deep learning models against phishing email attacks is proposed in this work. Various deep learning models were trained using the features obtained from the head and body parts of incoming e-mails in the proposed method. As a result of the tests, a 96.84% success rate was achieved with this detection method proposed against phishing attacks.

Keywords: *Social Engineering Attack, Phishing E-mail Detection, Deep learning, Multi Layer Perceptron , LSTM, Word2Vec.*

1. Giriş

Dünya üzerindeki internet aktivitesinin ve aktif kullanıcı sayısının artması, insanların birbirleriyle kolay bir şekilde bağlantı kurmalarına, birçok veriye erişebilmelerine, hizmet ve ürünlerini bu şekilde temin etmelerine neden olmuştur. Büyük bir para trafiğinin bu sanal topluluklarda dönmesi dolandırıcıların ve hırsızların da iştahını kabartmaktadır. Eski zamanlarda daha çok fiziksel aktivitelere dayanan hırsızlık ve dolandırıcılık yöntemlerinin sanal ortama aktarılması ile hırsızların işlerinin kolaylaşması ve kısa süre zarfında çok fazla olay gerçekleştirebilme imkanlarının olması bu alanı daha da güçlendirilmeye muhtaç bırakmaktadır. İnsanlara ait bu değerli bilgileri çalmanın yollarından bir tanesi de kimlik avı saldırılarıdır. Kimlik avı, bugün yapılan farklı (ve kazançlı) dolandırıcılık türlerinden biridir. Ceza hukukunda sahtekârlık, yalnızca kişisel kazanımların veya bireyin imajının sarsılmasına yönelik kasıtlı bir aldatma olarak tanımlanır. Genel anlamda sahtekârlık, finansal veya kişisel kazanımlar amacıyla insanları kendi kişisel bilgilerini ifşa etmelerini sağlamak ve onları kandırmak için bir dizi eylem olarak tanımlanabilir. Kimlik avı, yasal bir kuruluşun kopya web sitesini oluşturarak kullanıcılardan (genellikle hırsızlık amacıyla) hassas veya gizli bilgilerini elektronik olarak elde etmeye çalışan bir eylemdir. Kimlik avı genellikle bir elektronik aygıt (tablet ve bilgisayar gibi) ve bir bilgisayar ağı yardımıyla yapılır; son kullanıcıların (güvenlik zincirindeki en zayıf unsur olduğu düşünülen) çeşitli tespit sistemlerinde mevcut olan zayıflıkları hedef alınır [1, 2]. Kimlik avı saldırganları, hedeflenen kullanıcının hesabına yetkisiz erişim elde etmek amacıyla sistemde kullandığı kişisel bilgilerini açığa çıkarmaya ikna etmek için kullanıcılara yönelik titiz bir şekilde oluşturulmuş iletileri (sosyal mühendislik iletileri olarak bilinir) ileterek kötülüklerini sürdürürler. Örneğin, bir kullanıcıya gönderilen sahte bir e-posta bir kötü amaçlı yazılım (tarayıcıda man-in-the-browser (MITB) olarak adlandırılır) içerebilir; bu kötü amaçlı yazılım, web tarayıcı ActiveX bileşenleri, eklentileri veya e-posta ekleri şeklinde olabilir; Bu kullanıcı farkında olmadan bu eki bilgisayarına indirirse, kötü amaçlı yazılım kullanıcının bilgisayarına kendini yükler ve kullanıcı (yani, banka hesabının meşru sahibi) bir çevrimiçi işlem yapmaya çalıştığında bu bilgileri dolandırıcılara iletir [1]. Bu saldırıları durdurmak ve savunma mekanizmalarının kullanıcıların tekrar dolandırmalarını önlemek için daha güvenli hale getirilmeleri gerekir ki bu da mevcut savunma sistemlerinin (tasarımlarının ve teknolojisinin) büyük ölçüde geliştirilmesi gerektiği anlamına gelmektedir [3]. Behdad ve ark [3] savunma sisteminin iyileştirilmesinin sahtekârları durdurmak için yeterli olmadığını ve bazılarının hâlâ nüfuz edebileceğine dikkat çekti; sistem aynı zamanda hileli dolandırıcılık faaliyetlerini tanımlayabilmeli ve gerçekleşmesini engelleyebilmelidir. Bugün çeşitli e-posta filtreleri tarafından kullanılan birçok geleneksel yaklaşım doğası gereği statiktir; yeni ve ortaya çıkan kimlik avı modellerini kaldırabilecek kadar sağlam değildir; yalnızca mevcut kimlik avı kalıplarını işleme yetenekleri vardır, böylece e-posta kullanıcılarını yeni kimlik avı saldırılarına karşı savunmasız bırakmaktadırlar. Bu bir kısır döngü olarak devam etmektedir, çünkü dolandırıcıların faaliyetlerini sürekli hale getirme istekleri; tespit edilmeyi önlemek amacıyla çalışma biçimlerini mümkün olduğunca sık değiştirmektedirler. Bu, birçok araştırmacıyı hem bilinen hem de ortaya çıkan sahtekarlığı ele alabilecek diğer etkili teknikler aramaya motive etti ve bu da makine öğrenme algoritmalarının keşfedilmesine yol açmıştır. Bu makine öğrenme algoritmaları, sınıflandırma amacıyla kullanılan bir veri kümesinden yeni veya mevcut kalıpları (özellikleri) keşfetmek için veri madenciliği yöntemini kullanan yapay zekanın bir alt dalıdır. Derin öğrenme ise makine öğrenmesi algoritmalarında kullanılan yöntemlerin katmanlar haline getirilerek her katmanın birbiri arasında bilgi paylaşımını yapmasını sağlayan çalışma alanıdır. [4]

Bu çalışmada, kimlik avı e-postalarından kaynaklanan sosyal mühendislik saldırılarından etkilenen kullanıcı sayısını azaltmak ve kimlik avı e-postalarını sınıflandırmak amacıyla karma derin öğrenme modelleriyle oluşturulmuş bir sınıflandırıcı yöntem önerilmiştir. Önerilen bu sınıflandırıcıda karma derin öğrenme modellerinin eğitilmesi için özellik çıkarılabilecek iki temel kısımdan oluşturulmuştur: E-postadaki, url sayısı veya e-postadaki ek sayısı gibi e-postanın özelliklerini barındıran başlık kısmı ve e-postanın alıcısı tarafından okunması amaçlanan e-posta metnini içeren e-posta gövde kısmı.

Yapılan çalışmada, 4512 e-postadan oluşan bir veri kümesinden 11 önemli kimlik avı özelliği

(literatürden tanımlanmıştır) çıkarılmıştır [5] ve bu özellik çıkarımından sonra, her e-posta için, bu özelliklerin bir vektör temsili oluşturulmuş olup, daha sonra modeli eğitmek için kullanılmıştır. İkinci kısımda bulunan e-posta metni ise çeşitli işlemlerden geçirilerek vektör haline getirilmiş ve farklı bir modeli eğitmek için kullanılmıştır. Daha sonra bu 2 modelden çıkarılan veriler farklı bir özellik matrisinde birleştirilerek son karar verici modele iletilmiş ve sınıflandırma işlemi tamamlanmıştır.

Makalenin geri kalan bölümleri şu şekilde düzenlenmiştir: Bölüm 2 önerilen yöntem hakkında arka plan bilgisi sağlarken, bu alandaki ilgili çalışmaları özetlemektedir. Sonraki 3. Bölümde önerilen yaklaşım açıklanırken, Bölüm 4'te değerlendirme ve deney sonuçları verilmiştir. Son Bölüm sonuç ve önerilerdir.

2. İlgili Çalışmalar

Bu bölüm, kimlik avı e-postalarının sınıflandırılması için araştırmacılar tarafından yapılan önceki çalışmaların kısa bir sunumunu içermektedir.

Zareapoor ve ark. [8] tarafından, Kimlik avı e-posta tespiti üzerine bir vaka çalışması ile sınıflandırma performansı açısından özellik boyut azaltma yönteminin en iyi sonuçları verdiği bir test yapılmıştır. Bu deneylerin tamamlandığı sınıflandırma kimlik avı e-posta sınıflandırmasıdır. Kullanılan sınıflandırıcı, nihai bir tahmin oluşturmak için rastgele oluşturulmuş eğitim setleri tarafından eğitilmiş sınıflandırıcıların kombinasyonu olan bir torbalama sınıflandırıcısıdır. Temel sınıflandırıcı olarak J48 karar ağacı algoritması kullanılmıştır.

Park ve Taylor [12] kimlik avı e-postası için sözdizimsel özellikleri kullanarak yasal e-postalar arasında fiillerin öznesi ve nesnesinin kullanımlarında ki karşılaştırılmasını bildirmektedir. Kullanılan e-posta veri kümeleri 2005 ve 2014'te toplanan kimlik avı ve güvenli e-posta kümeleridir. İki farklı kimlik avı e-posta veri kümesi kullanmanın amacı, yıllar boyunca kimlik avı e-postaları arasında gelişen farklılıklar üzerine karşılaştırma yapmaktı. Bu çalışma, kimlik avı ile yasal e-postalar arasında aynı fiillerin öznesinde ve nesnelerinde bir fark olup olmadığını bulmaya çalışmaktadır. Kimlik avı sınıflandırması için sözcüklerin kendileriyle ilgili araştırmalar yapılmış olsa da bu daha fazla tahmin ve daha karmaşık bir sınıflandırma sağlamaktadır.

Almomani ve ark [6] kimlik avı e-postasının çeşitli sınıflandırma ve değerlendirme yöntemleri ile kimlik avı e-postasının temel özellikleri, gizli konu modeli özellikleri, dinamik Markov Zinciri özellikleri gibi farklı özellikleri üzerine tartışılmıştır. Ağ düzeyinde koruma, kimlik doğrulama tekniği, istemci tarafı araçları ve filtreleri, kullanıcı eğitimi ve sunucu tarafı filtreleri ve sınıflandırıcı gibi kimlik avı e-postalarına karşı çeşitli koruma önlemlerine ışık tutulmuştur. Kimlik avı e-posta tespiti için mevcut çeşitli makine öğrenme yaklaşımları tartışılmıştır. Bu çalışmada sunulan ve değerlendirilen yaklaşımlar, sözcük modeli, çoklu sınıflandırma algoritması, sınıflandırıcı model tabanlı özellikler, kimlik avı e-postasının kümeleme yaklaşımları, çok katmanlı sistemler ve kimlik avı e-postalarını algılamak ve sınıflandırmak için gelişen bağlantı sistemi temelli yöntemlerdir. Gelecekteki çalışmalar olarak, çevrimiçi olarak çalışabilecek ve sıfır gün kimlik avı e-posta algılama ile ilişkili sınırlamaları etkili bir şekilde çözebilecek yeni bir yaklaşım geliştirmeyi önerdiler. Silva ve ark [7] mevcut çeşitli makine öğrenme algoritmalarını sunmuş ve değerlendirmiştir. Çalışma [7] içerik tabanlı özelliklere, bağlantıya dayalı özelliklere ve dönüştürülmüş bağlantıya dayalı özelliklere göre web sitelerini güvenli veya zararlı olarak sınıflandırmaya odaklanmıştır. Deney için WEBSPAM UK2006 toplama veri kümesini kullandılar. Monte Carlo çapraz doğrulamasını, eğitim ve test alt kümelerinin boyutunu tanımlamak için kullanmışlardır. Tüm sınıflandırıcılar arasında karar ağaçları ve Adaboost gibi toplama teknikleri en iyi sonucu verirken SVM en kötü sonuçları vermiştir.

Hussain ve Qamar [27] kimlik avı e-postalarını tespit etmek için bir topluluk modeli (çoğunluk oylaması) önermektedir. Bu çalışmada kimlik avı e-postalarını değil, yaramaz (spam) olarak işaretlenen e-postaları sınıflandırmaya çalışsa da çoğunluk oylama yöntemi geliştirdiğimiz yönteme paralel bir işleyiş içerisinde olması incelemeye değer bir çalışma olduğunu göstermektedir. Çalışmada kullanılan sınıflandırma algoritmaları, Saf Bayes, Destek Vektör Makineleri, Rastgele Orman, Karar ağaçları ve K-NN En Yakın Komşu algoritmalarıdır. Çoğunluk oylaması önce birkaç basit sınıflandırıcıyı eğiterek ve daha sonra nihai sınıflandırmayı yapmak amacıyla tüm sınıflandırıcıların çıktılarını karşılaştırarak çalışır. Kullanılan örnek veri seti UCI'den [31] alınmıştır.

Modelleri oluşturmak ve eğitmek için RapidMiner kullanılmıştır.

Kimlik avı koruması, metinsel veri havuzundaki kimlik avı içeriğini / belgelerini algılamayı amaçlar. Doğal dil işleme (NLP), akıllı tanıma gerçekleştirmek için metin içeriğini analiz edebildiği için bu sorun için doğal bir çözüm sunar. Bu çalışmada, e-postalar için kimlik avı önleme sorununu gidermek amacıyla NLP' de metin kategorizasyonu (yani bir e-postanın kimlik avı olup olmadığını tahmin etme) kullanılmıştır. Bu teknikler, son zamanlarda topluluktan büyük ilgi gören derin öğrenme modellerine dayanmaktadır. Özellikle, e-postaları aynı anda sözcük ve cümle düzeyinde modellemek için hiyerarşik uzun kısa vadeli bellek ağları (H-LSTM) ve dikkat mekanizmalarına sahip bir çerçeve sunulmuştur.[9]

Basnet ve ark [11] kimlik avı e-postalarına temel yapısal özellikler dahil ederek ve sınıflandırma süreci için veri setine farklı makine öğrenme algoritmaları kullanarak kimlik avı e-postalarını sınıflandırmaktır. Belirli bir eğitim setinden makine öğrenmesinin kullanımı, örneklerin etiketlerini (kimlik avı veya yasal e-postalar) öğrenmektir. Bu çalışma, kimlik avı e-postalarının sınıflandırılması amacıyla farklı makine öğrenme algoritmalarının kullanılmasının etkinliği hakkında bilgi vermektedir.[11] Moradpoor ve diğerleri [32], Kimlik Avı E-postalarının Tespiti ve Sınıflandırılması için önişleme aşamaları sözcük gömme veya vektörleştirme olmak üzere sinir ağı tabanlı bir sınıflandırıcı önermiştir. Model altı bölümden oluşur ve beş özellik kullanır. Eğitim süreci için on kat çapraz doğrulama yapıldı. Kullanılan e-posta veri setleri, yasal e-postalar için Spam Assassin'den [33] ve Jose Navario kimlik avı e-posta kuruluşundan [20] alınmıştır.

3. Önerilen Yöntem

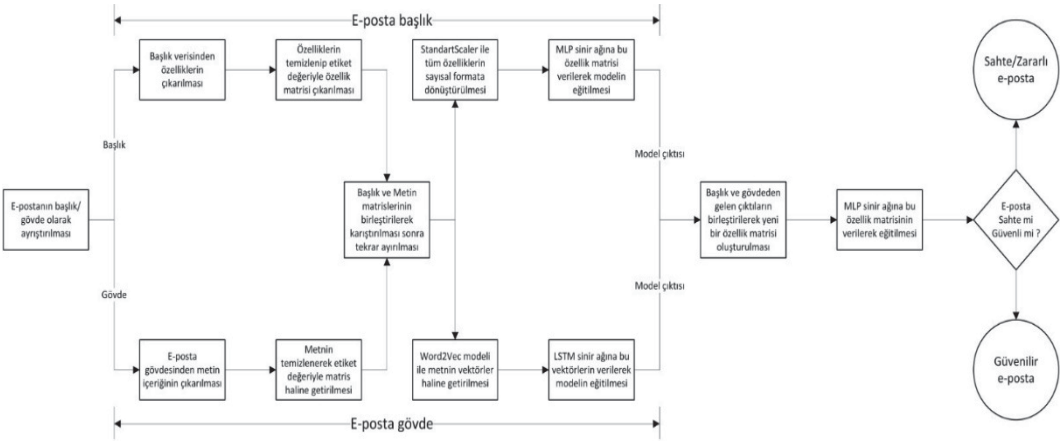
3.1. Genel Bakış

Kimlik avı e-posta saldırılarını tespit edebilmek için geliştirilen bu yöntemde alınan e-postalar, incelenmesi için 2 parçaya bölünmüştür. Bu parçalar gelen e-postanın başlık ve gövde kısımlarıdır. E-postalar başlık kısmında gönderici, alıcı, içerik tipi, konu, ek dosyalar gibi özellikleri barındırır. Gövde kısmında ise genel mesaj içeriği bulunmaktadır. Fakat gönderici bu mesaj içeriğine resimler, dosyalar ve bağlantı adresleri ekleyebilir. Mesaj içeriğinde olan bu özelliklerde başlık analizi olarak adlandırılacak ilk analiz modeline gönderilmiştir. Bu sayede metin işleme yalnızca bir modelde gerçekleştirilirken, eposta içerisinden çıkarılan diğer tüm özellikler başlık analizi içerisinde toplanmıştır. Geliştirilen yöntem e-postanın başlık ve gövde bölümlerini ayrı olarak incelemiş, daha sonra her iki kısım için oluşturulan özellik matrislerine göre farklı derin öğrenme modellerine göndererek bu modeller eğitilmiştir. Başlık değerlerine göre oluşturulan özellik matrisi Çok katmanlı algılayıcı (Multilayer perceptron-MLP) modeline girdi olarak verilmiştir. Gövdeden çıkarılmış metin değerleri ise temizlenip word2vec modeli ile vektörler haline getirilerek LSTM sinir ağına girdi olarak verilmiştir. Bu iki modelden alınan çıktılar işlenerek yeni bir özellik matrisine dönüştürülerek, karar verecek olan yine tamamen bağlı (fully connected-FC) çok katmanlı algılayıcı bir modele verilmiştir. Bu çok katmanlı model ile diğer iki modelden gelen işlenmiş sonuçlar üzerinden daha doğru ve keskin tahminler yapılmaya çalışılmıştır. Yalnızca metin veya yalnızca başlık verilerinin kullanılmamasının sebebi 2 özellik kümesinin de birbirini destekleyip tek bir modelin veriyi ezberlemesinin önüne geçmektir. Şekil-1'de sistemin genel yapısı gösterilmiştir.

3.2. Analiz

3.2.1. Başlık Analizi

E-postalar başlık kısımlarında gönderici ve niyeti hakkında belirgin izler taşıyabilirler. Oluşturulan yöntem e-postanın kimlik avı niyetiyle gönderilip gönderilmediğini kontrol etmek amacıyla geliştirilmiştir. Bu sebeple eposta başlığında çıkarılacak doğru ve yol gösterici öznitelikler e-postanın yüksek doğrulukla sınıflandırılması için çok önemlidir. Yöntem dahilinde daha önceki çalışmalardan da edinilen istatistikler doğrultusunda, kimlik avı e-postalarının en çok oranda barındırdığı ve tespit için en belirgin özellikler baz alınarak bir özellik seti oluşturulmuştur. Bu özellik seti ve özellikleri Çizelge 1 ve devamında verilmiştir.



Şekil- 1: Bu çalışmada geliştirilen karma modelin mimarisi

Çizelge-1: Başlık kısmından çıkarılan özellik kümesi

Özellik No	Özellik Adı
1	URL içerisinde @ işareti bulunması
2	E-postada bulunan eklerin sayısı
3	CSS kontrolü
4	Harici kaynaklar
5	Flash içeriği kontrolü
6	HTML içeriği kontrolü
7	HTML Form kontrolü
8	HTML iFrame kontrolü
9	URL de IP kontrolü
10	Javascript kontrolü
11	URL sayısı

1. URL içerisinde "@" işareti bulunması

E-posta metni içerisinde veya başlık içerisinde bulunan tüm URL'lerin "@" işareti barındırıp barındırmadığını kontrol eden özellik. Eğer kontrol edilen URL'ler içerisinde "@" işareti barındırıyorsa doğru olarak, bulundurmuyorsa yanlış olarak işaretlenmiştir.

2. E-postada bulunan eklerin sayısı

E-postalar ile ek dosya gönderimi yapılabilmektedir. Bu dosyalar zararlı yazılım içerebilirler. Bu özelliğe gönderilen e-postanın kaç adet ek dosya içerdiği eklenmiştir.

3. CSS kontrolü

E-postalar yalnızca düz metinler olarak gönderilmemektedir. Zararlı e-postalar da kullanıcıyı aldatmak amacı ile genellikle görsel e-postalar hazırlanmaktadır. Bu hazır e-posta kalıpları içerisinde CSS stil dosyaları barındırmaktadır. Bu özelliğe e-posta içerisinde bulunan stil dosyalarının sayısı eklenmiştir.

4. Harici kaynaklar

E-posta metni içerisinde bulunan CSS stil ve Javascript etiketleri harici bir kaynağı referans gösterebilir. CSS stil dosyaları için kullanılan "style" etiketi "href" özelliği ile, Javascript dosyaları için kullanılan "script" etiketi ise "src" özelliği ile harici kaynakları referans gösterir. Bu özelliğe e-posta metni içerisinde yukarıdaki şekilde belirtilen yöntem ile referans gösterilen harici kaynakların sayısı eklenmiştir.

5. Flash içeriği kontrolü

E-postalar içerisinde flash içerikler bulunabilmektedir. Geçmiş zamanlarda birçok kez flash dosyalar ile dağıtılan zararlı yazılımlar ve flash dosyaların barındırdığı zafiyetler üzerine çalışmalar yapılmıştır. Bu özellik e-posta içerisinde flash içeriği bulundurup bulundurmadığını kontrol etmektedir, eğer bulunduruyorsa doğru olarak, bulundurmuyorsa yanlış olarak işaretlenmiştir.

6. HTML içeriği kontrolü

E-postalar yalnızca düz metin olarak gönderilmemektedir. Kimlik avı dolandırıcıları kullanıcıları aldatmak için görsel e-posta içerikleri hazırlayabilmektedir. Bu hazırlanan içerikler ile bazı kurum ve kuruluşların taklit edilmiş içerikleri ile kullanıcılar aldatılabilir. Bu özellik e-posta içerisinde HTML içeriği bulundurup bulundurmadığını kontrol etmektedir, eğer bulunduruyorsa doğru olarak, bulundurmuyorsa yanlış olarak işaretlenmiştir.

7. HTML Form kontrolü

E-postalar içerisinde bulunan HTML içerikleri formlar barındırabilmektedir. Bu formlar ile kimlik avı dolandırıcıları kullanıcılardan aksiyonlar alarak bilgilerini ele geçirmeye çalışırlar. Bu özellik e-posta içerisindeki HTML içeriğinin form etiketi bulundurup bulundurmadığını kontrol etmektedir, eğer bulunduruyorsa doğru olarak, bulundurmuyorsa yanlış olarak işaretlenmiştir.

8. HTML iFrame kontrolü

iFrame, HTML içeriğinde belirli bir çerçeve içerisine farklı bir internet sayfasını çağırıp, görüntülenmesine yardımcı olan bir HTML etiketidir. Bu özellik e-posta içerisinde bulunan HTML içeriğinin iFrame etiketini barındırıp barındırmadığının kontrol edilmesi ile doğru ya da yanlış olarak işaretlenmiştir.

9. URL de IP kontrolü

E-posta içerisinde bulunan URL'lerin etki alan adı yerine IP adresi barındırabilmektedir. Kimlik avı dolandırıcıları etki alan adı yerine IP adresi ekleyebilmektedir. Bu özellik e-postaların etki alan adı yerine IP adresi barındırıp barındırmadığının kontrol edilmesi ile doğru ya da yanlış olarak işaretlenmiştir.

10. Javascript kontrolü

E-postalar kullanıcıyı aldatmak amaçlı çeşitli Javascript betikleri kullanabilirler. Bu betikler ile kullanıcıların birtakım aksiyonlar almalarını sağlayabilirler. Bu yüzden e-postaların içerisinde ki Javascript betiklerinin sayısı önemlidir. Bu özellikte e-posta içerisindeki script etiketlerinin sayısı eklenmiştir.

11. URL sayısı

Bu özellik e-posta içerisinde bulunan toplam aktif URL sayısını kontrol etmektedir. Tespit edilen bu özellikler veri kümesi içerisindeki tüm e-postalar için çıkartılmış ve özellik matrisi oluşturulmuş olup, Çizelge-2'de görselleştirilmiştir. Bu özellikler yalnızca e-postaların başlıklarında bulunan özellikler değildir, gövdesinde bulunan özellikler de buraya eklenmiştir. Çünkü e-posta gövdesinde bulunan metinler gövde analizi başlığı altında anlatılacağı şekilde sözcük vektörleri haline getirilerek LSTM sinir ağı ile eğitilmiştir. Başlık analizinde çıkarılan özellikler, bu haliyle derin öğrenme yöntemine gönderilmeden önce bir dizi ön işleme tabi tutulmuştur.

Çizelge-2: Başlık analizi sonrası oluşan özellik matrisi gösterimi

	@inURLs	Attachments	Css	ExternalResources	Flashcontent	HTMLcontent	HtmlForm	HtmliFrame	IPsinURLs	Javascript	URLs
4147	False	0	0	0	False	True	False	False	True	0	47
4186	False	0	0	0	False	True	False	False	True	0	8
2958	False	0	0	0	False	True	False	False	False	0	2
2738	False	0	0	0	False	True	False	False	False	0	2
4210	False	0	0	0	False	True	False	False	False	1	7
...	...	...	...	...	...	...	...	...	...	...	...
2081	False	0	0	0	False	False	False	False	False	0	0
239	False	0	0	0	False	False	False	False	False	0	0
3826	False	0	0	0	False	True	False	False	False	0	10
3873	False	0	2	2	False	True	False	False	True	12	43
2778	False	0	0	0	False	True	False	False	False	0	2

4512 rows × 11 columns

Bu işlemde standart ölçekleyici kullanılmıştır. Standart ölçekleyici standart skor hesaplama yöntemini kullanmaktadır. Bu ölçekleyici özellik matrisindeki değerlerin standart bir değere (puan) dönüştürülmesini sağlamaktadır.

Dönüşüm sonrasında ortaya çıkacak bu standart puan, ölçeklenecek değerden diğer tüm değerlerin ortalamasının çıkarılıp, tüm değerlerin standart sapmasına bölünmesiyle ifade (1) ile hesaplanmaktadır: [13]

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Burada;

μ değerlerin ortalamasını,

x standart puana dönüştürülecek değeri,

σ ise değerlerin standart sapmasını temsil etmektedir. Mutlak z değeri ise standart sapma birimlerinde dönüştürülecek değer ile ortalama değer arasındaki mesafeyi temsil etmektedir. z değeri dönüştürülecek değer ortalamanın altında ise negatif, üstünde ise pozitif olacaktır. [13, 14]

İfade (1)'de tanımlanan standart ölçekleyici sayesinde Çizelge-2'de gösterilen e-posta içerisinden çıkarılan 11 adet özelliğin standart puan yöntemine göre dönüştürülmesi sonucunda yeni bir özellik matrisi elde edilmiştir. Çizelge-2'de görünen doğru ve yanlış olarak işaretlenen verilere ise önce, doğru = 1, yanlış = 0 tam sayı dönüşümü yapılmıştır. Standart ölçekleyici dönüşümü sonrasında oluşan özellik matrisinden bir kesit Şekil-2'te gösterilmiştir.

```
[[-0.02105847, -0.059655  , -0.10824129, ...,  2.69232778,
  -0.0998656 ,  3.12055807],
 [-0.02105847, -0.059655  , -0.10824129, ...,  2.69232778,
  -0.0998656 ,  0.15387414],
 [-0.02105847, -0.059655  , -0.10824129, ..., -0.3714258 ,
  -0.0998656 , -0.30253878],
 ...,
 [-0.02105847, -0.059655  , -0.10824129, ..., -0.3714258 ,
  -0.0998656 ,  0.30601178],
 [-0.02105847, -0.059655  ,  4.58776554, ...,  2.69232778,
   17.92387874,  2.8162828 ],
 [-0.02105847, -0.059655  , -0.10824129, ..., -0.3714258 ,
  -0.0998656 , -0.30253878]])
```

Şekil-2: Ölçeklendirilmiş veri kümesi

3.2.2. Gövde Analizi

Başlık analizi bölümünde e-posta içerisindeki sınıflandırmayı en çok etkileyen özellik kümesi oluşturulmuştur. Gövde analizi içerisinde bulunan özellikler de başlık analizinde oluşturulan özellik matrisine eklenebilmesi için başlık analizi bölümüne gönderilmiştir. Gövde analizinde ise gelen e-postanın metin içeriği kısmı üzerinden bir özellik matrisi çıkarılmıştır. Öncelikle e-postanın metin değerinin içerisinden eğer varsa bütün html etiketleri temizlenmiştir. Html etiketleri oluşturulacak sözcük vektörlerini olumsuz yönde etkilediği ve çok sık tekrarlar gerçekleştirdikleri için temizlenmeleri gerekmektedir. Bu html etiket temizliğini Python dili içerisinde bulunan "BeautifulSoup" kütüphanesi ile yapılmıştır. [30] Metin içeriği html etiketlerinden arındırıldıktan sonra içerisinde URL bulundurup bulundurmadığı kontrolü yapılmıştır. Bulunan URL'ler başlık analizindeki özelliklere eklendikten sonra metin içerisinden çıkarılmıştır. Metin içerisinde ki gereksiz boşluklar ve tüm sayılar Python dili ve regex yapısı ile temizlenmiştir. Bu temizleme işlemlerinin sonucunda metin içerisinde hiçbir sayı, özel karakter, tanımlanamayan ifade, URL ve html etiketi bırakılmamıştır. Saf metin olarak kalan değerle küçük harflere dönüştürüldükten sonra, özellik matrisi oluşturulmak üzere word2vec olarak adlandırılan modele verilmiştir. Word2vec verilen metin içerisinde ki karakter veya sözcükleri vektör uzayında ifade etmek için vektörlere dönüştüren, denetimsiz tahmin etmeye dayalı bir modeldir. Google tarafından 2013 yılında geliştirilmiştir. 2 tür yöntem kullanmaktadır. Sürekli sözcük torbası (Continous bag-of-words) ve skip-gram mimarilerini verimli bir şekilde uygulanmasını sağlar [15]. Yüksek boyutlu bir vektörün, düşük boyutlu bir alana dönüştürülmesi işlemine yerleştirme (embedding) işlemi denilmektedir. Bu Word2vec modeli ile yapılan işleme ise sözcük yerleştirme (word embedding) denilmektedir. Metin kalıplarının vektör haline getirilmesi için keras kütüphanesinde [16] bulunan Tokenizer sınıfı ile metin değerleri önce sözcük sözcük ayrılarak belirteç (token) haline getirilmiştir. Tokenize işleminden sonra tam sayı dizilerine dönüştürülmüştür. Bu da keras'ta bulunan text_to_sequence yöntemi ile gerçekleştirilmiştir. Metinlerden alınan sözcük sayısı 300 ile sınırlandırılmıştır. Bu sayede tüm e-postalardan eşit uzunlukta sözcük vektörleri oluşturulmuştur. Şekil-3' te vektör haline getirilmiş sözcük dizilerinden bir kesit gösterilmiştir.

```
array([[   1,  290,  129, ...,  522,  291, 2891],
       [   0,    0,    0, ...,  488, 5444,  776],
       [   0,    0,    0, ...,  311, 5446, 3545],
       ...,
       [   0,    0,    0, ...,    7, 1835, 1261],
       [2350,    8, 2356, ..., 1255,  797,  272],
       [   0,    0,    0, ...,   10,  311, 1301]], dtype=int32)
```

Şekil-3: Vektör haline getirilmiş sözcük dizileri

Son olarak ise Google tarafından 3 milyon sözcük ile eğitilmiş Word2Vec modelinden [15] alınan vektörler ile, Şekil-3'te gösterilen dizileri sabit boyutlu yoğun vektörlere dönüştürme işlemi için yerleştirme katmanı (embedding layer) oluşturulmuştur.

3.3. Derin Öğrenme Modeli

Günümüzde üretim sanayi ve elektronik sektörünün hızla gelişmesi nedeniyle eski yıllara göre sorunlar ve hesaplama problemleri katlanarak büyümüştür. Bu büyümeler aynı zamanda ortaya çok fazla verinin çıkmasına neden olmuştur. Süregelen bu süreçte çok miktarda verinin işlenmesi hatta eski veri yığınlarının içerisinden gelecek bir durum veya olay tahmini yapılması mümkün hale gelmiştir. Makine öğrenmesi bu büyük veri yığınlarının içerisinden belirli problemleri çözmek veya belirli durumlar için tahminlerde bulunabilmek için ortaya çıkmıştır. Derin öğrenme ise bir makine öğrenmesi yöntemidir, belirleyici farkı ise katmanlar halinde çalışması ve bu katmanların birbirleri arasında bilgi aktarımı olmasıdır. Yani herhangi bir katman kendinden önceki katmanlardan bilgiler alarak kendi bilgisini revize edebilir. Verilen bir veri kümesiyle çıktıları tahmin edebilmek için farklı öğrenme yöntemleri kullanılabilir. Farklı öğrenme türleri takip eden satırlarda verilmiştir.

Denetimli Öğrenme

Denetimli bir öğrenme algoritması, önceden sınıflandırılmış örneklerden istenen modeli eğitir ve öğrenir. Algoritma, bir hatayı hesaplamak ve parametreleri uyarlamak için örneği giriş olarak ve her örneğin önceden belirlenmiş çıktısını kılavuz olarak kullanır. Böyle bir sınıflandırıcının performansını değerlendirmenin yaygın bir yolu, sınıflandırılmış veri setini iki bölüme ayırmaktır: sınıflandırıcıyı eğitmek için bir kısım (eğitim kümesi) ve bir performansını değerlendirme kısmı (test kümesi). İlk olarak sınıflandırıcı eğitim seti kullanılarak eğitilir ve daha sonra sınıflandırıcı önceden belirlenmiş sınıflandırmalarıyla ilgili bilgileri kullanmadan test kümesi örneklerini sınıflandırmaya çalışır. Bir sonraki adım, performans değerlendirme amacıyla sınıflandırıcının tahminlerini gerçek belirlenmiş sınıflandırmalarla karşılaştırmaktır. Denetimli öğrenme algoritmaları sınıflandırma ve regresyon algoritmalarına ayrılır. Sınıflandırma algoritmaları söz konusu olduğunda, bunların amacı her örnek için önceden belirlenmiş bir kümeden bir değer seçmektir. Örneğin, bir e-postayı kimlik avı veya güvenli olarak sınıflandırma durumunda, sınıflandırma algoritması güvenli bir e-posta tahmini için '0' ve kimlik avı e-posta tahmini için '1' değerini verir. Regresyon algoritmaları, çıktılar bir aralık içinde sayısal bir değere sahip olduğunda kullanılır.

Denetimsiz Öğrenme

Denetimsiz algoritmalar, sınıflandırılmamış bir dizi girdi örneği alır ve bunları bir araya getirerek veya veri noktalarının komşularına kümeleyerek bir model bulmaya çalışır. Bu algoritmaların amacı, sağlanan verilere ilişkin bilgi sağlamak için, örneğin herhangi bir kümede gruplandırılamayan aykırı değerleri vurgulayarak veya belirli eğilimleri bağlı olarak göstererek, işlemin başında bilinmeyen bilgi veya modelleri bulmaktır. Bu çalışma ile e-postaların kimlik avı veya güvenli olarak sınıflandırılmasıdır. Bu bir sınıflandırma problemidir ve denetimli öğrenme kullanılır. Bu, yeni sınıflandırılmamış e-postaları sınıflandırabilmek için bir modeli eğitmek için kullanılacak önceden sınıflandırılmış bir veri kümesi olacağı anlamına gelir. Denetimli öğrenmede çıktıları tahmin etmek için bir modelin eğitilmesi için girişler sağlanır. Bir model, derin öğrenme sürecinden çıkan sınıflandırıcıdır ve her örneği y tahminiyle eşleştirir. Temel olarak derin öğrenme modelinin çalışma prensibi doğrusal regresyon örneği üzerinden verilmiştir. Bu modelde, n özellik sayısı, x veri setinde ki bir örneği ifade eden değişken, tüm örnekler için bu hesaplama yapılmaktadır. Her örnek için sağlanan etiketler/çıkışlar vardır ve temel model şu formüle sahiptir:

$$y' = b + w_1x_1 + w_2x_2 + \ \dots \ + w_nx_n \qquad (2)$$

(2)'de b önyargı (bias)'dır ve w her örnek için bir ağırlık değeridir. Denetimli öğrenmede algoritma, her örnek geçtikten sonra önyargı ve ağırlıkların değerlerini güncelleyerek bir model oluşturur, böylece belirli bir kayıp fonksiyonu kullanılarak

tahmin hatası en aza indirilir. Deneylerde kullanılan kayıp fonksiyonu, her tahmin ve o tahminin gerçek değeri arasındaki ortalama kare farkını hesaplayan ortalama kare hatasıdır (3).

$$MSE = \frac{1}{n}\sum(y - \hat{y})^2 \quad (3)$$

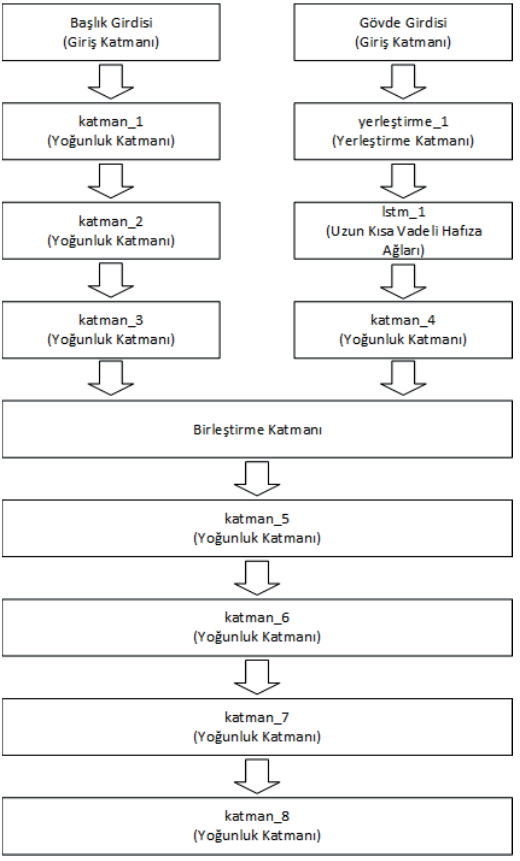
Aktivasyon fonksiyonları ise her nöronda hesaplanan değerin çıktısını istenilen aralığa sıkıştırmaya veya istenilen şekilde almak için kullanılan fonksiyonlardır. Deneylerde kullanılan aktivasyon fonksiyonları ise ReLu (Rectified linear unit) ve Sigmoid fonksiyonlarıdır. ReLu (4)'deki gibi ifade edilmektedir:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4)$$

(4) ile gelen negatif değerleri 0 değerine sabitlerken diğer değerleri aynı bırakmaktadır. Bu aktivasyon fonksiyonunu kullanma nedenimiz gövde analizinden yalnızca pozitif değerlerin gelmesi, modelin dengesiz karar vermesine neden olmaktadır. Bu yüzden başlık analizindeki negatif değerleri (4) fonksiyonu ile dengeleyip Sigmoid fonksiyonu (5) ile 0 ve 1 aralığına dönüştürülmüştür. Bu işleme normalizasyon işlemi denilmektedir. (5) fonksiyonu ile verilen değerler 0 ve 1 arasına normalize edilmiştir. Modelimiz tahmin sonucunda kimlik avı veya güvenli olarak işaretlemek istediği için bir sınıflandırma yapması gerekmektedir. Sonuçlarımız 0 veya 1 olduğu için sigmoid aktivasyon fonksiyonu bu işlemi gerçekleştirmiştir. Matematiksel tanımı (5)'deki şekildedir:

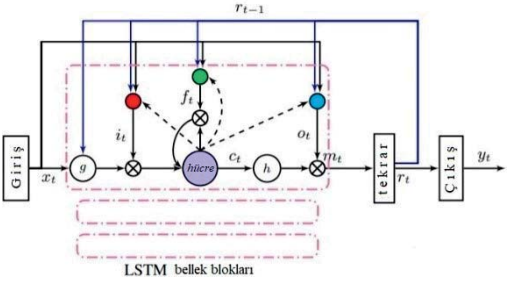
$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Son olarak model eğitiminde her adım da güncellenen değerler ile birlikte tahmin edilen değerin gerçek değere daha fazla yaklaşabilmesi için bir takım optimizasyon işlemlerinden geçmesi gerekmektedir. Bu çalışmada Adam optimizer [17] kullanılmıştır. Adam optimizer ile değişkenler optimize edilmiştir. Gövde analizinde oluşturulan yerleştirme katmanından sonra kullanılacak olan Uzun Kısa Vadeli Bellek (Long Short Term Memory - LSTM) ağları 1997 yılında tanıtılmıştır. LSTM mimarisi giriş, unutma ve çıkış olmak üzere 3 kapı, blok girişi, Sabit Hata Döngüsü, çıkış aktivasyon fonksiyonu ve gözetleme (peephole) bağlantılarına sahiptir [19]. Bloğun çıktısı tekrar tekrar bloğun girişine ve tüm kapılarına bağlanır.



Şekil-4: Derin öğrenme modelinin görselleştirilmesi

Gözetleme bağlantıları ve unutma kapısı ilk geliştirilen mimaride bulunmamaktadır. LSTM'in kendi durumunu sıfırlamak için unutma kapısı [18], kesin zamanlamaları öğrenmeyi kolaylaştırmak için ise gözetleme bağlantıları eklenmiştir [18].



Şekil-5: LSTM mimarisi. [18]

Başlık analizinde oluşturulan özellik matrisi ilk oluşturulan modele verilmeden önce eğitim ve test verisi olarak ayrılmıştır. Toplam 4512 veri içerisinden %80 (3609) eğitim, %20 (903) test verisi olacak şekilde bölündükten sonra, modele girdi olarak verilmiştir. Başlık analizi modelinde ilk katmanda 6 adet nöron bulunmaktadır, ardından gelen gizli yoğunluk katmanında da 6 adet nöron bulunmaktadır ve bu katmanda (4) fonksiyonu kullanılmıştır çünkü standart ölçekleme işleminden sonra oluşan özellik matrisi negatif değerler içerebilmektedir bu negatif değerleri optimize edebilmek için bu aktivasyon fonksiyonu kullanılmıştır. Daha sonra son çıkış katmanında 1 adet nöron bulunup aktivasyon olarak (5) fonksiyonu kullanılmıştır. Bu sayede çıktı 0 ve 1 arasında ölçeklendirilmiştir. Gövde analizi modelinde vektör dizilerine dönüştürülen ve ardından özellik matrisi haline getirilen veriler öncelikle yerleştirilme katmanına gönderilmiştir. Bu katmanda 300 adet nöron bulunmaktadır. Fakat bu katmandaki değişkenler eğitilemez değişkenlerdir ve sadece yerleştirme işleminde kullanılmıştır. Ardından LSTM katmanı gelmektedir ve bu katmanda 128 adet nöron bulunmaktadır ve onu, (5) fonksiyonu kullanan bir çıkış katmanı takip etmektedir. Bu iki analiz modelinden çıkan değerler birleştirilerek 3 adet sırayla 128,64,32 nörona sahip gizli yoğunluk katmanlarına ardından (5) fonksiyonu kullanan bir çıkış katmanına sahip modele bağlanmıştır. Bu modelde iki özellik grubu için tek derin öğrenme yöntemi kullanmak yerine, birden fazla model kullanılmasının nedeni sayısal ve niteliksel değerlerin, metin değerleriyle birlikte eğitilerek düşük ve optimal olmayan sonuçlara neden olacağı düşünülmesidir. Bunun yerine sayısal değerler için en iyi sonuçları alabileceğimiz modeller, metin değerleri için ise sözcük vektörleri ve LSTM modellerinin kullanımı alınan sonuçları iyileştirilmiştir. İkinci bir neden ise yalnızca başlık analizi veya yalnızca gövde analizi ile yapılacak hatalar veya sınıflandıramama problemini birbirlerini destekleyerek çözmeleri beklenmiştir. Geliştirilen bu derin öğrenme modelinin özeti Çizelge-3'de gösterilmiştir.

Çizelge-3: Derin öğrenme modeli özeti

Katman (tür)	Çıktı	Değişken
Başlık Analizi		
Yoğunluk katmanı	6	78
Yoğunluk katmanı	6	42
Yapay sinir ağı	1	7
Gövde Analizi		
Yerleştirme katmanı	300	900000000
LSTM	128	219648
Yapay sinir ağı	1	129
Karar Katmanı		
Yoğunluk katmanı	128	384
Yoğunluk katmanı	64	8256
Yoğunluk katmanı	32	2080
Yapay sinir ağı	1	33
Eğitilen Toplam Değişken Sayısı:		230,657

4. Deney Sonuçları ve Değerlendirme

4.1. E-posta Veri Kümeleri

Bir derin öğrenme sınıflandırıcısı oluşturmanın ilk adımı uygun bir veri kümesine sahip olmaktır. Bu çalışmada yapılan araştırmalardan sonra aşağıdaki veri kümeleri bulunmuştur:

Jose Nazario kimlik avı e-postası veri kümesi [20]: Bu veri kümesi mbox biçimindeki dosyalarda kimlik avı e-postalarını içermektedir. E-postalar, veri kümesinin yaratıcısı tarafından sınıflandırılmıştır ve kendisine gönderilen e-postalardır. Literatürde birçok çalışmada kimlik avı posta kümesi olarak kullanılan bir veri kümesidir.

Enron Email Veri Kümesi [21]: Enron, dünyanın en büyük entegre doğal gaz ve elektrik şirketlerinden biriydi [22]. 2001 yılında iflas etmeden önce Enron, 2000 yılında yaklaşık 101 milyar dolar gelir ile 29.000 çalışan istihdam etti. 2001 yılı sonunda Enron'un raporlanan finansal durumunun, o zamandan beri Enron skandalı olarak bilinen sistematik ve planlanmış muhasebe sahtekarlığı tarafından sürdürüldüğü ortaya çıktı. Skandal nedeniyle, şirketin tüm e-postaları söz konusu skandalı araştırması sırasında Federal Enerji Düzenleme Komisyonu tarafından halka duyuruldu. Bu veri seti, çoğunlukla Enron'un üst düzey yönetimi olan ve klasörlerde düzenlenmiş yaklaşık 150 kullanıcının verilerini içerir. Şirket toplamda

yaklaşık 500.000 e-posta içeriyor. Bu e-postalar, literatür tarafından makine öğrenimi algoritması eğitimi ve testi amacıyla meşru bir e-posta veri kümesi olarak kullanılmıştır. Deneylerde kullanılan veri kümesi bu 2 kaynaktan derlenmiştir ve 2256 güvenli ve 2256 kimlik avı e-postası olarak ayrılmıştır. Toplamda 4512 e-postadan oluşmaktadır. Çizelge-4'te Veri kümesi görselleştirilmiştir.

Çizelge-4: Veri kümesinin özeti

Veri kümesi	güvenli	kimlik avı	Toplam
Eğitim	1,801	1,808	3,609
Test	455	448	903
Toplam	2,256	2,256	4,512

4.2. Değerlendirme ve Deney Sonuçları

Test kümesini daha önce elde ettiğimiz modele girecek ve bir dizi gösterge elde edeceğiz. Bu göstergeler, modelin performansını özel olarak değerlendirmek için kullanılır. Bu göstergeleri şöyle açıklayabiliriz [23]:

True Positive (TP): Doğru algılanan kimlik avı e-postaların sayısı.

False Negative (FN): Kimlik avı olan fakat güvenli olarak algılanan e-postaların sayısı.

False Positive (FP): Güvenli olan fakat kimlik avı olarak algılanan e-postaların sayısı.

True Negative (TN): Doğru algılanan güvenli e-postaların sayısı.

Çizelge-5'te sınıflandırma karışıklık matrisini gösterilmiştir.

Çizelge-5: Sınıflandırma karmaşıklık matrisi

Tahmin / Gerçek	0 (güvenli)	1 (kimlik avı)
0 (güvenli)	TN	FP
1 (kimlik avı)	FN	TP

Tüm modeli test etmek için aşağıdaki değerlendirme ölçütlerini göz önünde bulunduruyoruz:

Doğruluk (Accuracy)
Doğruluk, modelin veri kümesindeki gerçek sınıflandırmalarla karşılaştırıldığında elde ettiği doğru tahminlerin oranıdır. Şu şekilde hesaplanır:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (6)$$

Tutturma (Precision)
Kesinlik, doğru tahmin edilen pozitif örneklerin toplam pozitif tahminlere oranıdır:

$$Tutturma = \frac{TP}{TP+FP} \quad (7)$$

Bulma (Recall)
Hatırlama veya algılama olasılığı (Recall), doğru tahmin edilen kimlik avı e-postalarının toplam kimlik avı e-postaları miktarına oranıdır:

$$Bulma = \frac{TP}{TP+FN} \quad (8)$$

F-Değeri (F-measure)
F-skoru, hassasiyet ve hatırlamanın harmonik ortalamasıdır. F-skoru 0 ile 1 arasında bir değer aralığına sahiptir, burada 1 mükemmel hassasiyete eşittir ve her ikisi de 1'e eşittir. F1 skoru, hassasiyet ve hatırlama arasında bir ilişki metriği sağlamak için Yararlıdır. Çünkü bu iki değer bir şekilde ters benzetmeye sahiptir. Aşağıda tarif edilen ROC eğrisi ile temsil edilmektedir:

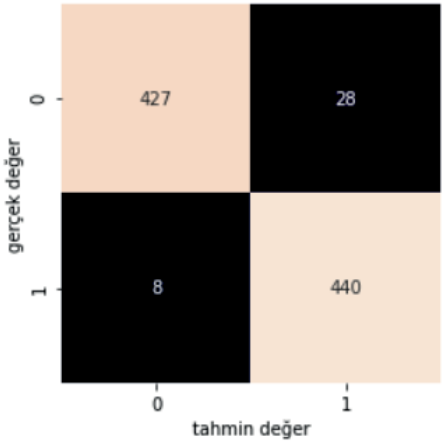
$$F - değeri = 2 * \frac{precision*recall}{precision+recall} \quad (9)$$

Yanlış alarm olasılığı (FPR)
Yanlış alarm olasılığı (FPR), pozitif bir tahminin yanlış olma olasılığıdır:

$$FPR = \frac{FP}{FP+TN} \quad (10)$$

Karışıklık Matrisi

Bir karışıklık matrisi, tüm olası tahmin ve gerçek sonuç kombinasyonlarının listelendiği 2x2'lik bir matristir. Karışıklık matrisinin her bir hücresinin değerleri, bir test setinde yürütülen ilgili tahminlerin sayısına eşittir. Test işlemlerinden sonra modelimizin karmaşıklık matrisi Şekil-6'da gösterilmiştir.



Şekil-6: Karışıklık matrisi

Yukarıda tanımlanan değerlendirme ölçütleri ise Çizelge-6'te gösterilmiştir. Bu ölçütlerin hesaplanmasında ise oluşturulan modelin sınıflandırılma raporu özelliğinden faydalanılmıştır. Bu özellik sklearn [24] kütüphanesinde bulunmaktadır.

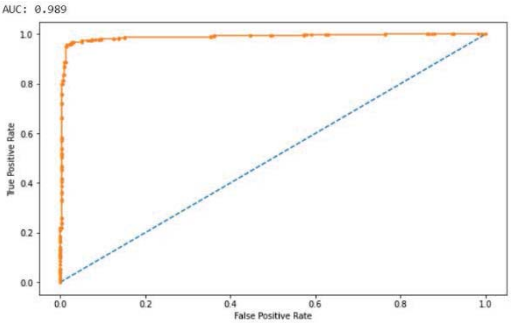
Çizelge-6: Sınıflandırma raporu

#	Precision	Recall	F-score	Accuracy
0	0.97	0.96	0.96	0.96
1	0.96	0.97	0.96	0.96

ROC-AUC Eğrisi:

ROC eğrisi, ikili sınıflandırma sistemlerinde ayrım eşik değerinin farklılık gösterdiği durumlarda, hassasiyetin kesinliğe olan oranıyla ortaya çıkmaktadır. ROC daha basit anlamda doğru pozitiflerin, yanlış pozitiflere olan kesri olarak da ifade edilebilir. Sınıflandırıcıların performans ölçümü için ROC analizi yapılarak doğruluk, hassasiyet ve kesinlik değerleri ile ROC puanı hesaplanabilir. [25]

Yapılan deneyler sonrasında edinilen ölçüt değerleriyle birlikte oluşturulan ROC-AUC eğrisi Şekil-7'de gösterilmiştir.



Şekil-7: ROC-AUC Eğrisi

Çizelge-7'da gösterilen yöntemler bu çalışmada kullanılan veri kümesinin birebir aynısını kullanmamışlardır. Fakat bu çalışmada kullanılan veri kümelerinin birçok varyasyonu bu yöntemlerde kullanılmaktadır. Veri kümesinin dengeli olması için 500.000 adet güvenli e-posta içerisinden sadece 2256 tanesi kullanılmıştır. Çünkü 2256 adet kimlik avı e-postası mevcuttur. Diğer yöntemlerde daha fazla veya az sayıda örnek kullanılmıştır. Ayrıca Çizelge-7'da farklı kamu kurum ve kuruluşlarından özel izinler ile alınan çeşitli kimlik avı e-postası veri kümeleri de kullanılmıştır.

Çizelge-7: Makine öğrenmesi algoritmaları ile sınıflandırma yapan diğer araştırmalar ile karşılaştırma

Referans	Yöntem	Doğruluk
[29]	SVM ,NaiveBayes, AdB	86
[10]	Fisher Ayrımsalık Analizi	93
[26]	Word embedding + CNN	94,2
[34]	Doc2vec+SVM	88,4
[28]	RNN	91
[35]	Multi-filter approach	92,72
[6]	SVM+NNet	97,99
[36]	NaiveBayes-DecisionTree	96,77
H-OLTA	Hibrit MLP - LSTM	96,84

Bu karşılaştırmalara ek olarak derin öğrenme yöntemlerinde yalnızca bu değerler üzerinden değerlendirme yapmak yeterli olmayacaktır. Aynı zaman da hesaplanan kayıp değeri de önemli bir değerlendirme ölçütüdür. Düşük kayıp oranları modelin ne kadar hassas tahminlerde bulunabildiğini göstermektedir. Bu sebeple modelin geliştirilirken hedeflenen başarı kriterlerinden bir tanesi de düşük kayıp değeridir. Yöntemimizde %3'lük bir kayıp

değeri oluşmaktadır ve bu kayıp değeri diğer birçok yönteme göre düşük bir seviyededir.

5. Sonuç ve Öneriler

Bu çalışmada, e-postaları kimlik avı ve düşük hata oranıyla güvenli olarak sınıflandırmak amacıyla derin öğrenme yöntemlerini birleştirerek hibrit bir yöntem olan 'H-OLTA' (Hibrit ve OLTAlama sözcüklerinin birleşiminden türetilmiştir.) yöntemi önerilmiştir. Kimlik avı e-posta tespit sistemlerinde doğruluk yüksek bir paya sahiptir. Geliştirilen yöntemlerin bir kısmı yalnızca metin bölümüyle ilgilenerek uzun süren metin işleme yöntemleri önermektedir. Bu yöntemlerin uzun sürme nedeni büyük sözcük havuzlarının verileri dönüştürme ve öğrenme süreleri oldukça uzun sürmesidir. Yalnızca özellik çıkarımı yöntemi metin değerlendirme yöntemlerine göre çok daha hızlı öğrenilebilse de tek başlarına tespit etmede hata oranları çok yüksek olabilmektedir. Bu yüzden geliştirilen H-OLTA yöntemi ile hem bu hata oranlarını düşürmek hem de kabul edilebilir hızlarda modeller eğiterek tahmin üretilmeye çalışılmıştır. Gövde analizi kısmında nispeten uzun süren tespit işlemini başlık analizinde kısa sürede eğitilen özellik çıkarımı modeli ile destekleyerek daha doğru ve tutarlı tespitlerde bulunulmuştur. Aynı zamanda H-OLTA yönteminde dengeli bir veri kümesi kullanılmıştır. Dengeli veri kümesi ile modelin eğitilmesi, modelin tek bir sınıfa eğilimli olmasını engellemek için önemlidir. H-OLTA yöntemi umut verici bir sonuç elde etmiştir. Özellik seçimi için yeni özellikler eklenebilir fakat gereksiz özellik eklenmesi başarı oranının düşmesi gibi sorunları da beraberinde getireceğinden özellik çıkarımı yapılırken tespit için en fazla etkiye sahip özellikler alınmalıdır. Deneysel sonuçlar daha fazla eğitim verisinin dahil edilmesinin kimlik avı tespit doğruluğunu iyileştireceğini gösterdiğinden gelecekteki çalışmalarda kimlik avı tespit sisteminin performansını daha da artırmak için eğitim verilerine daha fazla veri kümesi eklenebilir. Kimlik avı e-postalarını gerçek zamanlı olarak tespit etmek için kullanılan paralel stratejiler nedeniyle model yapım süresi önemli ölçüde azalsa da hesaplama süreleri tam anlamıyla minimize edilemediği için daha verimli paralel algoritmalar yazılabilir. Modern bir toplumda bilginin korunması son derece önemlidir ve bilginin etrafındaki güvenlik seviyesi sürekli olarak iyileştirilse de zayıf olan tek nokta manipülasyon tekniklerine duyarlı olan insan olmaya devam edecektir. Bu yüzden insanların bilinçlendirilmesi yapılması gereken en önemli çalışmadır.

Kaynakça

[1] Khonji, M. Iraqi Y., ve Jones, A. *Phishing detection: a literature survey,* IEEE Communications & Surveys Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[2] Sheng S., Holbrook, M. Kumaraguru, P. L. Cranor, F. ve Downs, J. *Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions,* in Proceedings of the 28th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI '10), pp. 373–382, Atlanta, Ga, USA, April 2010.

[3] Behdad M., Barone, L. Bennamoun, M. ve French, T. *Nature inspired techniques in the context of fraud detection,* IEEE Transactions on Systems, Man, and Cybernetics C: Applications and Reviews, vol. 42, no. 6, pp. 1273–1290, 2012.

[4] Akinyelu, A. A., ve Adewumi, A. O. *Classification of phishing email using random forest machine learning technique.* Journal of Applied Mathematics, vol. 2014, 2014.

[5] Mohammad, R. M., Thabtah, F., ve McCluskey, L. *Intelligent rule-based phishing websites classification.* IET Information Security, 8(3), 153-160. (2014).

[6] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., ve Almomani, E. *A survey of phishing email filtering techniques.* IEEE communications surveys & tutorials, 15(4), 2070-2090. (2013).

[7] Silva, R. M., Yamakami, A., ve Almeida, T. A. *An analysis of machine learning methods for spam host detection.* In 2012 IEEE 11th International Conference on Machine Learning and Applications, (Vol. 2, pp. 227-232). IEEE. (2012, December).

[8] Zareapoor, M., ve Seeja, K. R. *Feature extraction or feature selection for text classification: A case study on phishing email detection.* International Journal of Information Engineering and Electronic Business, 7(2), 60. (2015).

[9] Nguyen, M., Nguyen, T., ve Nguyen, T. H. *A deep learning model with hierarchical lstms and supervised attention for anti-phishing.* arXiv preprint arXiv:1805.01554. (2018).

[10] Özdemir, C., Ataş, M., ve Özer, A. B. *Classification of Turkish spam e-mails with artificial immune system.* In 21st Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE. (2013, April).

[11] Basnet, R., Mukkamala, S., ve Sung, A. H. *Detection of phishing attacks: A machine learning approach.* In Soft Computing Applications in Industry (pp. 373-383). Springer, Berlin, Heidelberg. (2008).

[12] Park, G., & Taylor, J. M. *Using syntactic features for phishing detection.* arXiv preprint arXiv:1506.00037. (2015).

[13] E. Kreyszig *Advanced Engineering Mathematics* (Fourth ed.). Wiley. p. 880, eq. 5. ISBN 0-471-02140-7. (1979).

[14] Spiegel, Murray R.; Stephens, Larry J *Schaum's Outlines Statistics* (Fourth ed.), McGraw Hill, ISBN 978-0-07-148584-5 (2008),

[15] *Tool for computing continuous distributed representations of words,* Google Jul 30, 2013, Accessed on: Nov. 2019. [Online]. Available: https://code.google.com/archive/p/word2vec/

[16] Chollet, F. Keras *Git Hubrepository.* [Online]. Available: https://github.com/fchollet/keras [Accessed 2020].

[17] Kingma, D. P., ve Ba, J. *Adam: A method for stochastic optimization.* ArXivpreprint arXiv:1412.6980. (2014).

[18] Şeker, A, Diri, B, Balık, H., *Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme.* Gazi Mühendislik Bilimleri Dergisi (GMBD), 3 (3), 47-64 (2017).

[19] Sak, H., Senior, A. W., ve Beaufays, F. *Long short-term memory recurrent neural network architectures for large scale acoustic modeling.* (2014).

[20] *Jose Navario phishing corpus,* [Online]. Available: https://monkey.org/~jose/phishing/. [Accessed 2020].

[21] W. W. Cohen, *Enron Email Dataset,* 8 May [Online]. Available: https://www.cs.cmu.edu/~enron/. [Accessed 2020].

[22] *Enron Corporation-Company Profile*, [Online]. Available: https://www.referenceforbusiness.com/history2/57/Enron-Corporation.html. [Accessed 2020].

[23] Vinayakumar, R., Barathi Ganesh, H. B., ve Kumar, M., ve Soman, K. P. *DeepAnti-PhishNet: applying deep neural networks for phishing email detection.* CEN-AISecurity@ IWSPA, 40-50. (2018).

[24] Kramer, O. Scikit-learn. *In Machine learning for evolution strategies* (pp. 45-53). Springer, Cham. (2016).

[25] T. Fawcett, *An Introduction to ROC Analysis*, Pattern Recognition Letters, vol. 27, Jun 2006, pp. 861-874.

[26] Abu-Nimeh, S., Nappa, D., Wang, X., ve Nair, S. October). *A comparison of machine learning techniques for phishing detection.* In Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit (pp. 60-69). (2007,

[27] Hussain, R., ve Qamar, U. *An Approach to Detect Spam Emails by Using Majority Voting.* In International Conference on Data Mining, Internet Computing and Big Data (BigData2014) (pp. 76-83). (2014).

[28] Das, A., ve Verma, R. *Automated email Generation for Targeted Attacks using Natural Language.* arXiv preprint arXiv:1908.06893. (2019).

[29] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., ve Almomani, E. *A survey of phishing email filtering techniques.* IEEE communications surveys & tutorials, 15(4), 2070-2090. (2013).

[30] Richardson, L. *Beautiful soup documentation.* April. (2007).

[31] Hopkins M., *UCI Machine Learning Repository, Spambase Data Set,* [Online].Available: https://archive.ics.uci.edu/ml/datasets/Spambase. [Accessed 2019]

[32] Moradpoor, N., Clavie, B., ve Buchanan, B. *Employing machine learning techniques for detection and classification of phishing emails.*

In 2017 Computing Conference (pp. 149-156). IEEE. (2017, July).

[33] *Spam Assassin spam email public corpus*, [Online].Available: https://spamassassin.apache.org/old/publiccorpus/. [Accessed 2020].

[34] Unnithan, N. A., Harikrishnan, N. B., Vinayakumar, R., Soman, K. P., & Sundarakrishna, S. *Detecting phishing* E-mail using machine learning techniques (2018).

[35] Sonowal, G., & Kuppusamy, K. S. *PhiDMA–A phishing detection model with multi-filter approach*. Journal of King Saud University-Computer and Information Sciences. (2017).

[36] Espinoza, B., Simba, J., Fuertes, W., Benavides, E., Andrade, R., & Toulkeridis, T. December). *Phishing Attack Detection: A Solution Based on the Typical Machine Learning Modeling Cycle.* In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 202-207). IEEE. (2019,