Pamukkale Univ Muh Bilim Derg, 24(5), 870-878, 2018

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması

Comparison of machine learning techniques for classification of phishing web sites

Tahir Emre KALAYCI[1*]

[1]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Manisa Celal Bayar Üniversitesi, Manisa, Türkiye.
tahir.kalayci@cbu.edu.tr

Geliş Tarihi/Received: 13.11.2017, Kabul Tarihi/Accepted: 18.02.2018
doi: 10.5505/pajes.2018.10846
* Yazışılan yazar/Corresponding author
Araştırma Makalesi/Research Article

Öz

Günümüzde makine öğrenmesi yöntemleri bilgisayarların daha doğru eylemler gerçekleştirmesi amacıyla birçok farklı şekilde kullanılmaktadır. Bu amaçla kullanıldıkları bir alan kimlik hırsızı web sitelerinin tespit edilmesidir. Kimlik hırsızlığı, önemli kişisel bilgileri çalmak amacıyla güvenilir web sitelerini taklit eden sahte web sitelerinin oluşturulduğu çevrimiçi bir saldırı biçimidir. Bu tehlikeyi gerçekleşmeden önlemek amacıyla web sitelerinin farklı özelliklere dayanarak kimlik hırsızı bir site olup olmadığının belirlenmesi önemlidir. Bu çalışmada, bir web sitesinin kimlik hırsızı olup olmadığını tahmin etmek amacıyla AdaBoost, çok katmanlı algılayıcı, destek vektör makinesi, karar ağacı, en yakın k komşu, Naïve Bayes ve rastgele orman makine öğrenmesi yöntemleri 9 farklı özellik içeren 1353 örnekten oluşan bir veri kümesinden yararlanarak karşılaştırılmıştır. Eğitim ve sınama şeklinde ikiye bölünmüş veri kümesiyle yapılan deneylerde karar ağaçlarından oluşturulan bir topluluk öğrenme yaklaşımı olan rastgele orman yöntemi, karşılaştırılan diğer yöntemlere göre daha başarılı olsa da çapraz doğrulamanın kullanıldığı durumda çok katmanlı algılayıcı daha yüksek bir başarım elde etmiştir.

Anahtar kelimeler: Makine öğrenmesi, Sınıflandırma, Kimlik hırsızlığı

Abstract

Today, machine learning approaches are used to make computers act more accurately for various purposes. In this manner, one area in which the machine learning approaches are used is the detection of phishing web sites. Phishing is an online threat, which depends on creating a fake web site that mimics a trustworthy web site to steal important personal information. It is important to predict whether a website is a phishing website in order to avoid this danger before it happens. In this study, AdaBoost, multilayer perceptron, support vector machine, decision tree, k-nearest neighbors, Naïve Bayes and random forest machine learning techniques are compared to predict the purpose of a website. This comparison is performed by experimenting over a dataset containing 1353 instances with 9 different features. The experimental evaluation is performed in two different settings. The first setting based on splitting the data into training and test sets. In this setting the evaluation results show that the random forest algorithm, which is an ensemble learning approach based on decision trees, outperforms other compared approaches. On the other hand, in the second setting based on cross validation, multilayer perceptron shows a better performance.

Keywords: Machine learning, Classification, Phishing

1 Giriş

Makine öğrenmesi, bilgisayarların gerçekleştirdikleri eylemleri daha doğru bir hale getirmek üzere değiştirmesi veya uyarlaması olarak tanımlanabilir [1]. Böylece bu eylemler, doğruluğun ilgili eylemlerin gerçek eylemleri ne kadar iyi temsil ettiğini ölçtüğü bir durumda, çok daha doğru bir duruma gelir [1]. Bir bilgisayar programı, bir T sınıfındaki görevler ve P başarım ölçütü bağlamında eğer T görevlerindeki P ile ölçülen başarımı E deneyimiyle birlikte artıyorsa, E deneyimiyle öğrenmiş demektir [2]. Makine öğrenmesini bilgisayarların bir başarım ölçütünü örnek veri veya geçmiş deneyim kullanarak eniyilemesi olarak da tanımlayabiliriz [3]. Bu kapsamda belirli parametrelere sahip bir model tanımlanarak bir bilgisayar programının bir eğitim verisi veya geçmiş deneyimleri kullanarak bu parametreleri iyileştirecek şekilde çalıştırılması da öğrenme sürecini oluşturur.

Makine Öğrenmesi Bilgisayar Bilimleri, Mühendislik ve İstatistik gibi farklı bilim dallarından beslenir ve sınıflandırma, regresyon, kümeleme, yoğunluk kestirimi, örüntü tanıma, uç değer tespiti, sıkıştırma, bilgi çıkarma, vb. gibi farklı görevleri yerine getirmeye çalışır [2]. Bu görevleri gözetimli öğrenme ve gözetimsiz öğrenme ana başlıklarına ayırabiliriz. Gözetimli öğrenmede girdi kümesinin yanında çıktı kümesi de verilirken, gözetimsiz öğrenmede bilgisayarın bu çıktı kümesi verilmeden öğrenmesi beklenir. Örneğin sınıflandırma ve regresyon problemleri gözetimli öğrenme örnekleridir, kümeleme ve yoğunluk kestirimi ise gözetimsiz öğrenme örnekleridir [4].

Kimlik hırsızlığı kullanıcı adı, parola, kimlik bilgileri, vb. önemli bilgilerin çalınması amacıyla güvenilir web sitelerinin taklit edilmesi şeklindeki çevrimiçi bir tehdit biçimidir [5],[6]. Bu amaçla kimlik hırsızlığına yönelik web siteleri oluşturulmakta ve üstelik bu web siteleri özgün sitelere oldukça fazla benzetilmektedir [5]. Son yıllarda bu şekildeki saldırıların çok fazla arttığı hem gözlemlenmekte hem de raporlarla ortaya konulmaktadır [7]. Dolayısıyla farklı amaçlarla gerçekleştirilen kimlik hırsızlığının yarattığı sorunlar da gittikçe artmaktadır. Bu nedenle kimlik hırsızlığı gerçekleştiren web siteleri konusunda kullanıcıları herhangi bir hırsızlık gerçekleşmeden uyarabilmek, böylesi bir hırsızlığın yaratabileceği zararların önüne geçmek açısından önemlidir. Başlıca iki yöntemle bu soruna çözüm aranmaktadır [5]:

- *Kara liste yaklaşımları* web sitesinin URL'sinin kara listeye alınmış daha önceki kimlik hırsızı web siteleriyle karşılaştırılmasına dayanmaktadır. En önemli eksikliği bütün kimlik hırsızı sitelerin bu listeye girmesinin oldukça zaman gerektirmesi ve yeni siteleri tespitinin bu anlamda zorlaşabilmesidir,

- *Arama/Sezgisel yaklaşımlarda* web sitelerini betimlemek için kullanılan birçok özellik toplanmakta ve bu özellikler kimlik hırsızı sitelerin tespit edilmesinde kullanılmaktadır. Bu yöntemin kara liste yöntemlerine göre avantajı yeni yaratılan kimlik hırsızı sitelerin gerçek zamanlı olarak tespit edilebilmesidir [8].

Dolayısıyla kimlik hırsızı web sitelerinin tespiti, bir sınıflandırma problemi olarak ele alınabilir [5],[9]. Sınıflandırma özellikleri verilmiş olan bir örneğin hangi sınıfa ait olduğunu tahmin etme görevidir [4]. Toplanan bütün tekil ölçümler özellik (öznitelik) olarak adlandırılmaktadır. Bu özellikler sayısal değerlere sahip olabildiği gibi kategorik değerler de içerebilir. Sınıflandırma bir gözetimli öğrenme görevi olup, X girdi kümesinden Y çıktı kümesinin hesaplanmasına yönelik bir fonksiyonun elde edilmesidir [3]. Daha sonra bu elde edilen fonksiyon önceden karşılaşılmamış bir örneğin özelliklerini kullanarak, bu verili örneğin hangi sınıfa ait olduğunu tahmin etmek amacıyla kullanılır.

Bu problemin çözümüne yönelik olarak daha önce kimlik hırsızı olup olmadığı kanıtlanmış web sitelerini bir eğitim verisi olarak kullanarak yeni sitelerin sınıfını (kimlik hırsızı, şüpheli, kimlik hırsızı değil gibi farklı sınıflardan biri olacak şekilde) tahmin edecek bir sınıflandırma yöntemi kullanılabilir. Bu makalede kimlik hırsızı web sitelerinin sınıflandırılması amacıyla kullanılabilecek mevcut bazı gözetimli makine öğrenmesi yöntemleri karşılaştırılmaktadır. Bu yöntemlerin başarımlarını karşılaştırmak amacıyla daha önce ilişkisel sınıflandırma veri madenciliği yöntemiyle [5] elde edilmiş olan veri kümesinden yararlanılmaktadır. Karşılaştırma tek bir başarım ölçütüyle sınırlandırılmamakta, farklı ölçütleri de dikkate almaktadır. Ayrıca geleneksel olarak eğitim ve sınama verileriyle yapılan deneylerin yanı sıra çapraz doğrulama başarımlarına ilişkin deneyler de yapılmaktadır. Böylece farklı çalışmalarda farklı veri kümeleri kullanılarak karşılaştırılmış olan yöntemlerin, aynı veri kümesi kullanılarak farklı başarım ölçütlerine göre karşılaştırıldığı bir çalışma ortaya çıkmaktadır.

Bu makale şu bölümlerden oluşmaktadır: 2. bölümde daha önce bu konuda yapılmış olan çalışmalar özetlenmektedir, 3. bölümde bu çalışma kapsamında karşılaştırılan makine öğrenmesi yöntemleri anlatılmaktadır, 4. bölümde deneylere ilişkin genel bilgi ve deneylerin sonuçları anlatılmaktadır. Son bölümde de karşılaştırma sonucunda elde edilen sonuçlar aktarılmaktadır.

2 İlgili çalışmalar

Literatürde kimlik hırsızı web sitelerinin tespitine yönelik birçok çalışma olsa da bu bölümde makine öğrenmesi yöntemlerine odaklanan ilgili çalışmalar ele alınmaktadır.

Abdelhamid ve diğ. [5],[9] ilişkisel sınıflandırma veri madenciliği yöntemini kullanarak web sitelerini üç farklı sınıfa ayıran bir yöntem önererek farklı kaynaklardan toplanan gerçek veriyle yöntemlerini sınamıştır. Yaptıkları deneylere göre önerdikleri çok etiketli sınıflandırıcıya dayanan ilişkisel sınıflandırıcı (MCAC) yöntemi diğer yöntemlerden daha yüksek bir doğruluk elde etmiştir.

Aburrous ve diğ. [10] sınıflandırma madenciliği tekniklerini kullanarak kimlik hırsızı web sitelerinin nasıl tespit edileceğini incelemiştir. Çalışmada telefon aracılığıyla ve web sitesi aracılığıyla kimlik hırsızlığına ilişkin iki örneği ele alınmıştır. Web sitelerinin sınıflandırılmasına yönelik olarak C4.5, JRip, PART, PRISM, CBA ve MCAR veri madenciliği yöntemlerini karşılaştırdıkları deneyler, MCAR yönteminin diğer yöntemlerden daha başarılı olduğunu göstermektedir.

Kaytan [11] ile Kaytan ve Hanbay [12] kimlik hırsızı web sitelerinin tespit edilmesine yönelik olarak yapay sinir ağı ve aşırı öğrenme makinesi modelini karşılaştırmaktadır. Yapılan deneylere göre aşırı öğrenme makinesi daha iyi bir başarım göstermiştir.

Kazemian ve Ahmed [13] kötü niyetli web sitelerinin tespitine yönelik olarak üç gözetimli (en yakın k komşu, destek vektör makinesi, Naïve Bayes sınıflandırıcı) ve iki gözetimsiz (k-means, afinite yayılması) makine öğrenmesi yöntemini karşılaştırmıştır. Gözetimli yöntemlerde %89'un üzerinde bir doğruluk elde edilmiştir.

Koşan ve diğ. [14] kimlik hırsızı web sitelerinin tespiti amacıyla C4.5, ID3, PRISM, RIPPER, Naïve Bayes, en yakın k-komşu, rastgele orman yöntemlerinin karşılaştırmalı bir analizini sunmaktadır. Yapılan çalışmada seçilen veri kümesi açısından rastgele orman ve ID3 yöntemleri doğruluk oranı en yüksek yöntemler olarak öne çıkmıştır.

Lakshmi ve Vijaya [15] kimlik hırsızı sitelerin tespiti için gözetimli öğrenme yöntemlerinden çok katmanlı algılayıcı, karar ağacı ve Naïve Bayes sınıflandırıcıyı başka bir veri kümesi yardımıyla karşılaştırmıştır. Deney sonuçlarına göre karar ağacı çok daha başarılı sonuçlar elde etmiştir.

Miyamoto ve diğ. [8] AdaBoost, bagging, destek vektör makinesi, sınıflandırma ve regresyon ağaçları, lojistik regresyon, rastgele orman, sinir ağları, Naïve Bayes ve Bayes eklemeli regresyon ağacı yöntemlerini kimlik hırsızı web sitelerinin tespiti için karşılaştırmıştır. Yaptıkları deneylere göre AdaBoost yöntemi en başarılı sonucu elde etmiştir.

Moghimi ve Varjani [16] kimlik hırsızı web sayfalarının tespiti için kullanılabilecek hem URL hem de içerikten elde edilen özellikler önerdikten sonra destek vektör makinesi yardımıyla bir sınıflandırma gerçekleştirmiştir. Kuralların ve özelliklerin bir analizi de sunulmuştur.

Mohammad ve diğ. [17] zeki bir kural tabanlı yöntem önermişlerdir. Önerilen yöntem web sitelerinden otomatik olarak özellik çıkararak sınıflandırma kurallarını da üretmektedir. Kural tabanlı sınıflandırma algoritmalarını karşılaştırmak amacıyla gerçekleştirdikleri deneylere göre C4.5 algoritması RIPPER, PRISM ve CBA yöntemlerine göre daha başarılı bir sonuç elde etmiştir.

Mohammad ve diğ. [18] kimlik hırsızı web sitelerinin tespiti için kendi kendini inşa eden yapay sinir ağı yöntemi önermişlerdir. Önerilen yöntem otomatik bir şekilde ağı oluşturduğu, gibi yüksek bir kestirim başarısı göstermektedir.

Nguyen ve Nguyen [19] sadece URL değil sayfa içeriğinden de yararlanarak makine öğrenmesi yöntemleriyle kimlik hırsızı tespiti yapmaktadır. Yapılan çalışmada URL ve içerikten elde ettikleri özellikler kullanılarak J48 karar ağacı, rastgele orman, destek vektör makinesi, Naïve Bayes ve sinir ağları yöntemleri karşılaştırılmıştır. Deney sonuçlarına göre en başarılı sınıflandırma sonucunu rastgele orman yöntemi elde etmiştir.

Sahoo ve diğ. [20] salt kimlik hırsızı web sitelerini tespit etmekle yetinmeyen, kötü niyetli URL (web sitesi) tespitine yönelik makine öğrenmesi yöntemlerinin ayrıntılı bir incelemesini gerçekleştirmektedir. Çalışmada kara liste, sezgisel ve makine öğrenmesi yöntemlerinin ilkelerinden ve kullanılabilecek özelliklerden söz edildikten sonra kapsamlı bir literatür araştırması sunulmuştur.

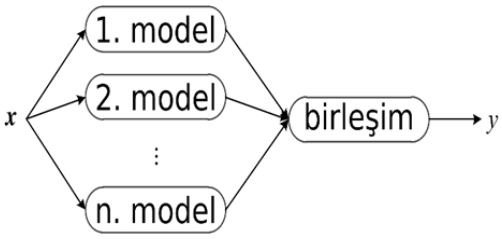
Bu çalışma, ilgili olanları yukarıda özetlenen literatürdeki çalışmalardan farklı olarak, birbirinden farklı veri kümeleri kullanılarak karşılaştırılan veya başka çalışmalarda başarılı sonuçlar üretmiş olan gözetimli makine öğrenmesi yöntemlerini tek bir veri kümesini kullanarak bir arada ve tek bir başarım ölçütüne bağlı kalmadan doğruluk, kesinlik, geriçağırım, F_1 ölçütü ve ROC AUC ölçütlerini kullanarak iki farklı deney kurulumunda (veri kümesini eğitim ve sınama parçalarına bölme ile çapraz doğrulama) karşılaştırmaktadır.

3 Karşılaştırılan yöntemler

Günümüzde farklı sınıflandırma problemleri için kullanılan yedi güncel makine öğrenmesi yönteminden web sitelerini sınıflandırmak amacıyla yararlanılmıştır. Bu yöntemler farklı kategorilere ait ve farklı özelliklere sahip olsalar da hepsinin ortak özelliği bir sınıflandırma probleminde kullanılabilmeleridir. Her bir yönteme ilişkin bilgi çok fazla ayrıntıya girmeden aşağıda verilmektedir, daha ayrıntılı bilgi için [1]-[4],[21],[22] kaynakları incelenebilir. Yöntemler, herhangi bir öncelik yanılsaması yaratmamak adına alfabetik olarak anlatılmaktadır.

3.1 AdaBoost

"Adaptive Boosting" (AdaBoost) yöntemi *Yoav Freund ve Robert Schapire* tarafından formüle edilen bir makine öğrenmesi meta-algoritmasıdır [23]. AdaBoost algoritması, bir topluluk öğrenme ("ensemble learning") yöntemidir [22] ve *boosting* yönteminin ilk pratik uygulamasıdır [24]. Topluluk yöntemleri, bir problemi çözmek için eğitim verisinden tek bir model oluşturan olağan öğrenme yöntemlerinden farklı olarak, ilgili problemi çözmek için birçok modelin eğitilmesine dayanmaktadır. Topluluk yöntemleri, eğitim verisini kullanarak birden fazla model oluşturur ve bu modellerin sonuçlarını sınıflandırma veya regresyon amacıyla birleştirirler (Şekil 1) [22].



Şekil 1: Topluluk öğrenme yönteminde birçok model oluşturulur, daha sonra bu modeller birleştirilir [22].

Boosting yöntemlerinde temel yaklaşım zayıf sınıflandırıcıların çıktılarının ağırlıklı bir toplam halinde bir araya getirilerek, istenen çıktının bu toplama dayanılarak oluşturulmasıdır. Böylece tek bir sınıflandırıcı değil de birden fazla sınıflandırıcı barındırır ve tahmin etmek için bu sınıflandırıcıların hepsinden yararlanır. Geleneksel olarak bu zayıf sınıflandırıcıları oluşturmak için *karar ağacı*ndan yararlanmaktadır.

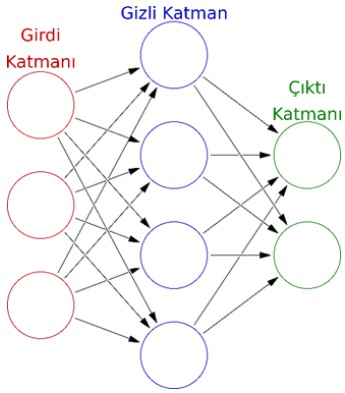
AdaBoost yönteminde her bir zayıf sınıflandırıcı, önceki zayıf sınıflandırıcı tarafından yanlış bir şekilde sınıflandırılmış eğitim örneklerine odaklanır ve onları iyi bir şekilde sınıflandırmaya çalışır. Böylece her bir sınıflandırıcı, yöntemin genel başarımına katkı sağlamaya çalışır.

$$H(x) = \sum_{1}^{T} \alpha_t h_t(x) \qquad (1)$$

AdaBoost yöntemi T farklı zayıf sınıflandırıcıyı eğitmekte, daha sonra da bu sınıflandırıcıların sonuçlarını alıp ağırlıklı oy çokluğuna dayanarak ilgili örneğin sınıfına karar vermektedir [24]. Buna ilişkin formül Denklem (1)'de gösterilmiştir. Formülde T eğitilen zayıf sınıflandırıcı sayısını, h her bir sınıflandırıcının elde ettiği sınıf sonucunu, α ilgili sınıflandırıcının ağırlığını ve H de karar verilen sınıfı belirtmektedir.

3.2 Çok katmanlı algılayıcı

Çok katmanlı algılayıcı (ÇKA) bir *ileri beslemeli yapay sinir ağı* türüdür. En azından üç katmandan (girdi katmanı, gizli katman, çıktı katmanı) oluşur (Şekil 2).



Şekil 2: Çok katmanlı algılayıcı bir girdi, bir çıktı ve bir veya daha fazla gizli katmandan oluşur.

Aşağıdaki üç nokta çok katmanlı algılayıcının temel özelliklerini vurgulamaktadır [21]:

- Ağdaki her bir nöron, lineer olmayan, *türevlenebilir* bir aktivasyon fonksiyonu içerir,
- Ağ hem girdi hem de çıktı katmanından *gizlenmiş* bir veya daha fazla ara katman içerir,
- Ağ, sinaptik ağırlıkların belirlediği boyutta yüksek bir bağlantısallık sergiler. Nöronlar girdilerinin ağırlıklı bir toplamını bulduktan sonra aktivasyon fonksiyonları yardımıyla çıktı üretirler. Denklem (2)'deki *Sigmoid* ve Denklem (3)'teki *hiperbolik tanjant* fonksiyonları yaygın olarak kullanılan aktivasyon fonksiyonlarıdır.

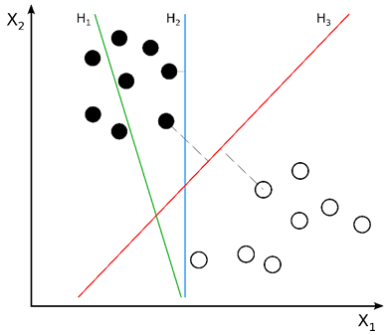
$$f(x) = \frac{1}{1 + e^{-x}} \qquad (2)$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (3)$$

Aktivasyon fonksiyonlarının türevlenebilir olması sayesinde, bütün ağ türevlenebilir bir fonksiyon olarak kabul edilerek, örneğin *Gradyan İniş* gibi yöntemlerle optimize edilebilir [22]. Çok katmanlı algılayıcıda ağın eğitilmesi amacıyla geri yayılım yöntemi kullanılmaktadır [25]. ÇKA, gözetimli öğrenme yöntemine göre çalışır, girdiler ve beklenen çıktılar ağa verilerek, ağdaki her bir düğümün ağırlıklarını öğrenmesi hedeflenir. İleriye doğru besleme ağın çıktısını hesaplarken, geriye doğru yayılım ortaya çıkan hatayı düğümlere yayarak ağın ağırlıklarını yeniden düzenlemesini sağlar, böylece hata oranının düşürülmesi amaçlanır. Bu süreç eğimin yönüne doğru uyumlama gerçekleştirilerek tamamlanır, bütün süreç hata minimize edilene veya işlem sonlandırılana kadar birçok kez tekrarlanır [22].

3.3 Destek vektör makinesi

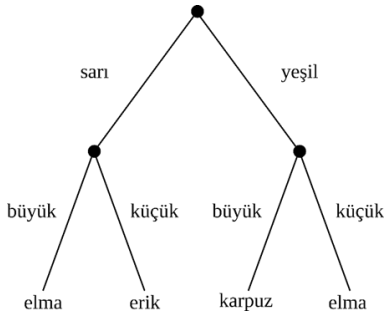
Destek vektör makinesi çok boyutlu bir uzayda herhangi bir girdi noktasından en uzakta yer alan en uygun lineer ayırıcı hiperdüzlemi bularak (Şekil 3) sınıflandırma gerçekleştiren bir yöntemdir [21],[27]. Şekil 3 incelendiğinde H_1 düzleminin iki sınıfı tam olarak ayırmadığı, H_2 düzleminin küçük bir uzaklıkla ayırdığı, H_3 düzleminin ise maksimum uzaklıkla ayırdığı görülmektedir. *Cortes ve Vapnik* [28] tarafından geliştirilen DVM veriye ilişkin herhangi bir birleşik dağılım fonksiyonu bilgisine ihtiyaç duymaz, dağılımdan bağımsız öğrenir [29]. Hem doğrusal hem de doğrusal olmayan problemler için kullanılabilmektedir. Doğrusal olmayan problemlerde öznitelik uzayını başka bir öznitelik uzayına dönüştürmek için kernel fonksiyonları kullanılır. Daha sonra dönüşümle elde edilen yeni öznitelik uzayında sınıfları birbirinden ayıran hiperdüzlem bulunur. DVM'lerin genelleme hatası ve hesaplama maliyeti düşüktür ve yorumlaması kolaydır, ancak uygun parametre ve kernel seçimine ihtiyaç duyarlar ve aslında özgün olarak yalnızca ikili sınıflandırma gerçekleştirirler [4].



Şekil 3: Destek vektör makinesi iki sınıfa ait olan noktaları ayıran bir düzlem bulmaya çalışır [26].

3.4 Karar ağacı

Karar ağacı böl ve yönet yaklaşımından yararlanan hiyerarşik bir gözetimli öğrenme modeli, parametre gerektirmeden verimli bir şekilde sınıflandırma ve regresyon gerçekleştiren bir yöntemdir [3]. Karar ağaçları basit ve anlaşılır bir yapıya sahiptir (Şekil 4).



Şekil 4: Meyveleri renk ve boyuta göre sınıflandıran basit bir karar ağacı [30].

Karar ağacı karar ve bitiş düğümlerinden oluşur. Her bir karar düğümü dalları etiketleyen ayrık bir test fonksiyonunu gerçekleştirmektedir. Verilmiş bir girdiye bu test fonksiyonu uygulanmakta ve sonucuna göre girdiye uygun bir dal seçilmektedir. Bu süreç kökte başlayıp, özyinelemeli olarak bir bitiş düğümüne ulaşana kadar sürdürülmektedir. Böylece girdinin ait olduğu sınıf tespit edilmektedir.

Sınıflandırma amaçlı kullanılan karar ağaçlarında dallara bölünme amacıyla farklı dallara ayırma ölçütlerinden yararlanılmaktadır. Günümüzde sıklıkla Denklem (4)'teki bilgi kazanımını ölçmek için kullanılan düzensizlik ("entropy") veya Denklem (5)'teki GINI katışıklığı ("impurity") fonksiyonları kullanılmaktadır [3]. Logaritmik fonksiyon hesaplaması gerektirmediği için GINI katışıklığı fonksiyonunun maliyeti daha düşüktür.

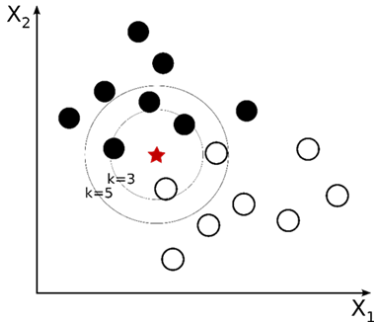
$$H(X) = -\sum_{x} P(x) log_2 P(x) \quad (4)$$

$$GINI(X) = 1 - \sum_{x} P(x)^2 \quad (5)$$

Sonuçların kolayca açıklanabilmesi, parametrik olmaması ve diğer yöntemlere göre görece daha hızlı olması nedenlerinden dolayı sınıflandırma ve kestirim problemleri için tercih edilmektedir [31]. Hesaplama açısından düşük maliyetlidir ve sonuçları da insanlar tarafından kolaylıkla anlaşılabilir, eksik değerlerle de ilgisiz özelliklerle de çalışabilir, ancak aşırı uyum sorunu yaratabilir [4]. Aşırı uyum sorununu çözmek için budamadan ve topluluk yöntemlerinden yararlanılmaktadır.

3.5 En yakın *k* komşu

En yakın k komşu parametrik olmayan bir sınıflandırma algoritmasıdır [32]. Herhangi bir girdinin sınıfı öznitelik uzayındaki hâlihazırda hangi sınıfa ait olduğu bilinen en yakın k komşusuna bakılarak belirlenmektedir (Şekil 5). Şekil incelendiğinde hem en yakın 3 komşuya hem de en yakın 5 komşuya göre yıldız ile gösterilen veri noktasının sınıfının siyah olduğu anlaşılmaktadır. Oldukça basit bir sınıflandırma modeli olduğu halde bazı problemlerde oldukça başarılı sonuçlar da vermektedir [25]. Veri hakkında bir varsayım gerektirmemesi, aykırı değerlere hassas olması ve yüksek doğruluk sağlaması avantajlarına sahip olsa da hesaplama anlamında maliyetlidir ve yüksek miktarda hafıza gerektirir [4].



Şekil 5: En yakın3 ve 5 komşuya göre değerlendirme için kullanılacak komşular ilgili daireler içerisinde gösterilmektedir.

3.6 Naïve Bayes sınıflandırıcı

Naïve Bayes olasılıktan yararlanarak sınıflandırma amacıyla kullanılan basit bir yöntemdir. *Bayesian karar kuramı*na dayanmaktadır ve temel olarak en yüksek olasılığa sahip kararın seçilmesi olarak özetlenebilir [4]. Az sayıda verinin olduğu durumlarda da çalışır ve çok fazla sınıfın yer aldığı problemleri de rahatlıkla çözebilir. En temel sıkıntısı girdi verisinin uygun şekilde hazırlanma gerekliliğidir.

Naïve Bayes yöntemi olasılık ve koşullu olasılıktan yararlanarak, ilgili girdinin ait olduğu sınıfı bulmaya çalışır. Bu amaçla Denklem (6)'daki (i sınıf sayısıdır) Bayes kuralından yararlanır [4].

$$p(c_i|x,y) = \frac{p(x,y|c_i)p(c_i)}{p(x,y)} \quad (6)$$

İki sınıflı örneği düşündüğümüzde aşağıdaki kural sınıfa karar vermek için kullanılır:

$$C(x) = \begin{cases} c_1, p(c_1|x,y) \geq p(c_2|x,y) \\ c_2, p(c_1|x,y) < p(c_2|x,y) \end{cases}$$

Bu kural ikiden fazla sınıf içeren problemlere rahatlıkla genişletilebilir.

3.7 Rastgele orman

Rastgele orman yöntemi, sınıflandırma ve regresyon amacıyla kullanılan bir başka topluluk öğrenme yöntemidir. Rastgele orman, her biri birbirinden bağımsız olarak ve aynı dağılım kullanılarak eğitim verisinden rastgele elde edilmiş bir örnekleme dayanan karar ağaçlarından oluşan bir topluluktur [33]. Bu yöntem eğitim sırasında birçok karar ağacı oluşturur ve daha sonra kestirim sırasında bu karar ağaçlarının sınıflandırma sonuçlarından yararlanılarak, girdinin sınıfına çoğunluk oyu aracılığıyla karar verilir. Rastgele ormanın en önemli avantajı, karar ağaçlarındaki aşırı uyum sorununa bir çözüm getirmiş olmasıdır. Bu yöntem basit olduğu gibi, kolayca da paralelleştirilebilir, AdaBoost yöntemine göre daha doğru sonuçlar ürettiği gibi, aşırı değer ve gürültülere daha dayanıklıdır, üstelik Mevcut bagging ve boosting yöntemlerinden daha hızlı çalışmaktadır [33].

4 Deneyler ve tartışma

Bu bölümde deneyler için kullanılan veri kümesi, deneylerin nasıl gerçekleştirildiği, karşılaştırma için kullanılan ölçütler ve deneyler sonucunda elde bulgular sunulmaktadır.

4.1 Veri kümesi

Deneyler için Abdelhamid ve diğ. [5] tarafından *ilişkili sınıflandırma veri madenciliği* yöntemiyle elde edilmiş olan *website phishing veri kümesi* [34] kullanılmıştır. Veri kümesinde *1353* örnek ve her bir örneğe ilişkin *9* özellik ve ait olduğu sınıf bilgisi yer alıyor. Her bir örneğin özellikleri istek URL'si, açılır pencere bilgisi, HTTPS ve SSL bilgisi, web trafiği, uzun URL olup olmaması, alan adının yaşı, IP adresi içerip içermediği, sunucu form işleyicisi (SFH) ve sayfa içi bağlantı bilgisinden yararlanarak oluşturulmuştur. Bütün özellikler ve örneklerin ait olduğu sınıf kategorik bir değer olarak kimlik hırsızı için *-1*, kimlik hırsızı olmayanlar için *1*, şüpheli olanlar için *0* içermektedir.

Veri kümesinin özellikleriyle ait olduğu sınıf değerine ilişkin yapılan korelasyon analizi sonuçları Tablo 1'de gösterilmektedir. Bu sonuçlara göre sınıf değerinin web trafiği özelliğiyle zayıf derecede pozitif bir ilişkisi vardır. SFH ile güçlü, açılır pencere ve SSL bilgisi özellikleriyle orta düzeyde negatif bir ilişkisi bulunmaktadır. URL uzunluğu ve IP adresi içermesi özellikleriyle çok zayıf düzeyde negatif bir ilişkiye sahipken, alan adının yaşı, istek URL'si, sayfa içi bağlantı özellikleriyle zayıf düzeyde negatif bir ilişkiye sahiptir.

4.2 Deneylerin gerçekleştirimi

Önceki bölümde anlatılan farklı yöntemlerle veri kümesindeki web siteleri sınıflandırılarak kimlik hırsızı olup olmadıkları tahmin edilmektedir. Böylece kimlik hırsızı web sitelerinin tespiti amacıyla, bu veri kümesi açısından aralarından en uygun yöntemlerin belirlenmesi amaçlanmaktadır. Deneyleri gerçekleştirmek için yaygın bir kullanım alanına sahip olan ve yukarıda sözü edilen farklı yöntemlerin gerçekleştirimlerini içeren scikit-learn kütüphanesinden [35] yararlanılmıştır. Bu kütüphane makine öğrenmesi alanında son zamanlarda artan bir şekilde kullanılan Python programlama dili ile geliştirilmiştir. Deneyler GNU/Linux işletim sisteminde, Python'un 3.6.3 sürümü, scikit-learn kütüphanesinin 0.19 sürümü kullanılarak gerçekleştirilmiştir.

Karşılaştırılan yöntemlerin gerçekleştirimleri için kütüphanenin aşağıdaki sınıflarından yararlanılmıştır:

- sklearn.ensemble.AdaBoostClassifier: AdaBoost sınıflandırıcı,
- sklearn.neural_network.MLPClassifier: çok katmanlı algılayıcı sınıflandırıcı,
- sklearn.svm.SVC: destek vektör makinesi sınıflandırıcı,
- sklearn.tree.DecisionTreeClassifier: karar ağacı sınıflandırıcı,
- sklearn.neighbors.KNeighborsClassifier: En yakın *k* komşu sınıflandırıcı,
- sklearn.naive_Bayes.GaussianNB: Gaussian Naïve Bayes sınıflandırıcı,
- sklearn.ensemble.RandomForestClassifier: rastgele orman sınıflandırıcı.

Tablo 1: Veri kümesinde örneğin ait olduğu sınıfı gösteren özellikle, diğer özellikler arasında yapılan korelasyon analizinin Pearson korelasyon katsayısı değerleri.

Özellik	Pearson Korelasyon Katsayısı
SFH	-0.678277
Popup Pencere	-0.509749
SSL Bilgisi	-0.518762
İstek URL'si	-0.271609
Sayfa içi bağlantı	-0.287007
Web trafiği	0.243896
URL uzunluğu	-0.183061
Alan adının yaşı	-0.231931
IP adresi içermesi	-0.059225

Deneylerde her bir sınıflandırma yöntemi için parametre değerleri olarak kütüphane içerisinde tanımlanmış olan varsayılan değerler kullanılmıştır. Parametrelere ilişkin en uygun değerlerin bulunması için gelecekte ek bir çalışma yapılması planlanmaktadır.

Deney amacıyla iki farklı yaklaşım sergilenmiştir. İlk yaklaşımda veri kümesi eğitim ve sınama şeklinde iki alt kümeye rastgele ayrılmıştır. Eğitim kümesi asıl veri kümesinin %70'ini (947), sınama kümesi de %30'unu (406) içermektedir. Daha sağlıklı bir sonuç elde etmek için deneyler 30 defa çalıştırılmış, sonuçların ortalaması sunulmuştur.

İkinci yaklaşımda kütüphanenin sunduğu çapraz doğrulama altyapısı yardımıyla *10*-katlamalı çapraz doğrulama [36] kullanılarak doğruluk değeri bulunmuştur. *10* katlamalı çapraz doğrulamada veri kümesi rastgele olarak *10* kümeye ayrılır, daha sonra *10* farklı aşamada sırasıyla bu kümelerden biri sınama veri kümesi, diğer kalan *9* küme de eğitim amacıyla kullanılır. Böylece her bir alt küme en az bir kez sınama amacıyla kullanılmış olur. Bu yaklaşımın en önemli avantajı aşırı uyum sorununa bir çözüm getirmesi ve böylece yöntemin daha iyi bir genelleme yapmasını sağlamasıdır.

4.3 Karşılaştırma ölçütleri

Sınıflandırma başarımını ölçmek için öncelikle hata matrisi kullanılmaktadır. Hata matrisi sınıf sayısı kadar satır ve

sütundan oluşan ve her bir hücrede farklı bir değer tutan bir tablodur (Tablo 2).

Tablo 2: İkili sınıflandırmaya ilişkin doğru (1) ve yanlış (0) sınıflandırma sonuç sayılarını gösteren bir hata matrisi.

		Tahmin Edilen Sınıf	
		0	1
Gerçek Sınıf	0	Doğru Negatif (DN)	Yanlış Negatif (YN)
	1	Yanlış Pozitif (YP)	Doğru Pozitif (DP)

Matrisin satırları gerçek sınıflandırma sonuçlarını gösterirken, sütunları tahminleri göstermektedir. Böylece hata matrisinden yararlanarak aşağıdaki ölçütler yöntemleri karşılaştırma amacıyla kullanılmaktadır [2]-[4]:

- **Doğruluk** ("accuracy"): Doğru olarak sınıflandırılmış girdilerin toplam girdilere oranını veren ölçüttür. Denklem (7) ile hesaplanmaktadır.

$$\frac{DP + YP}{DP + YP + DN + YN} \tag{7}$$

- **Kesinlik** ("precision"): Doğru olarak sınıflandırılmış pozitif girdilerin toplam pozitif değerlere oranıdır. Denklem (8) ile hesaplanmaktadır.

$$\frac{DP}{DP + YP} \tag{8}$$

- **Geriçağırım** ("recall"): Doğru olarak sınıflandırılmış pozitif girdilerin gerçek pozitif değerlere oranıdır. Denklem (9) ile hesaplanmaktadır.

$$\frac{DP}{DP + YN} \tag{9}$$

- **F_1 ölçütü**: Daha bir karşılaştırma sağlamak için kesinlik ve geriçağırımın harmonik ortalaması şeklinde hesaplanan bir ölçüttür (Denklem (10)).

$$\frac{2 * K * G}{K + G} \tag{10}$$

- **ROC AUC**: Yöntemler arasında tutarlı bir karşılaştırma yapabilmek için kullanılan bir başka ölçüt ROC ("Receiver Operator Characteristic") eğrisinin altında kalan alanın ("Area Under the Curve") hesaplanmasıyla elde edilmektedir. ROC eğrisi x ekseninde yanlış sınıflandırılmış pozitiflerin oranını (Denklem (11)), y ekseninde doğru sınıflandırılmış pozitiflerin oranını (Denklem (12)) gösteren ve sınıflandırma sonuçlarının yerleştirildiği bir grafiktir. Sınıflandırıcının bu değerleriyle ROC üzerinde bir noktadan geçen eğri oluşturularak altında kalan alan sınıflandırıcıları karşılaştırmak için kullanılır.

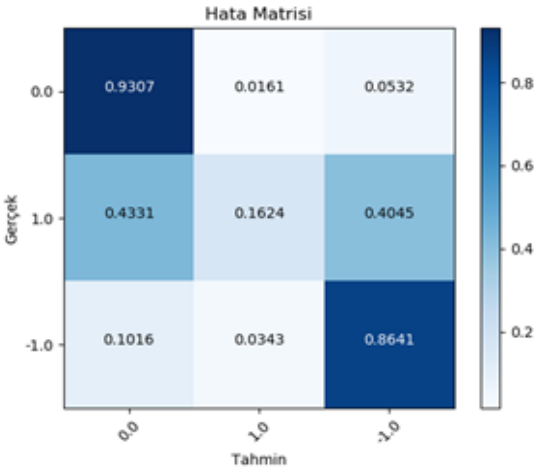
$$\frac{YP}{YP + DN} \tag{11}$$

$$\frac{DP}{DP + YP} \tag{12}$$

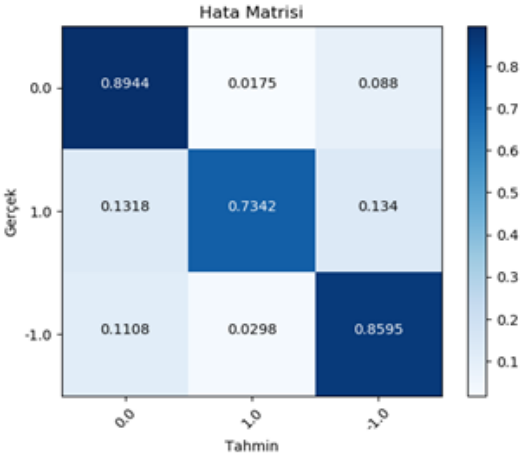
Kesinlik, geriçağırım, F ölçütü ve ROC AUC için ikiden fazla sınıf olması nedeniyle ağırlıklı hesaplama tercih edilmiştir. Bu hesaplamada her bir sınıf için değerler hesaplandıktan sonra bu değerlerin, sınıfa ait gerçek örneklerin sayısıyla ağırlıklandırılmış bir ortalaması alınmaktadır. Böylelikle hesaplama olası sınıf dengesizliğini dikkate almaktadır.

4.4 Bulgular

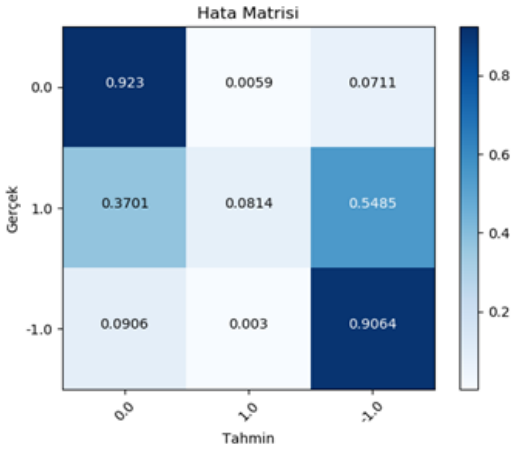
Bu bölümde ilk olarak web sitelerinin özelliklerini barındıran veri kümesi üzerinde işletilen farklı yöntemlerin hata matrisleri sırasıyla Şekil 6'da verilmiştir.



(a): AdaBoost.

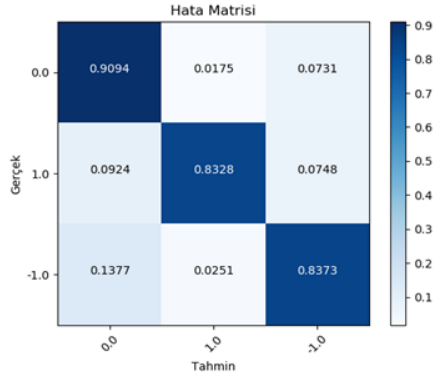


(b): Çok Katmanlı algılayıcı.

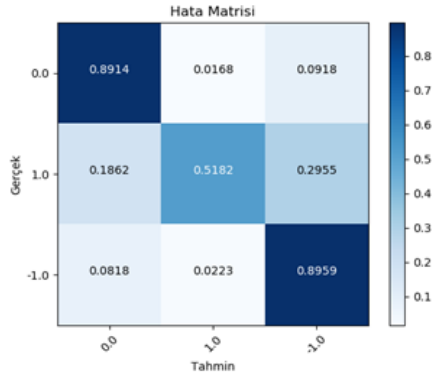


(c): Destek vektör makinesi.

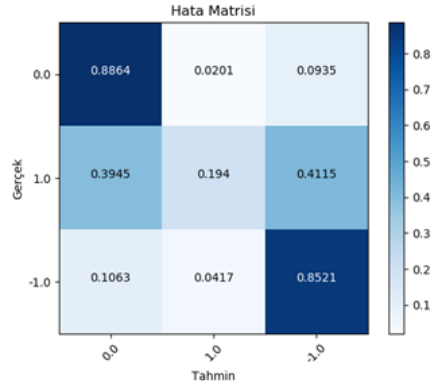
Şekil 6: Yöntemlerin bütün çalıştırılmaları sonucu elde edilen hata matrisleri (değerler normalize edilmiştir).



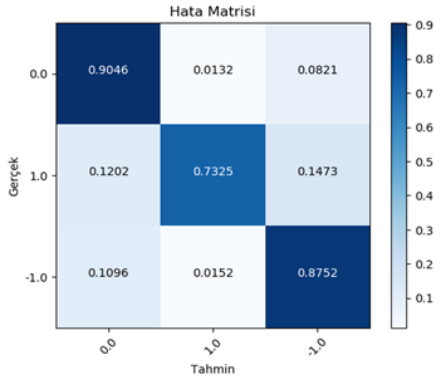
(d): Karar ağacı



(e): En yakın *k* komşu.



(f): Naïve Bayes.



(g): Rastgele orman.

Şekil 6: Yöntemlerin bütün çalıştırılmaları sonucu elde edilen hata matrisleri (değerler normalize edilmiştir).

Bu matrisler veri kümesinin %30'luk test kısmına ilişkin yapılan deneylerin sonuçlarını normalize edilmiş bir şekilde göstermektedir. Bu matrisleri incelediğimizde en doğru sonuçları *0 için AdaBoost, -1* için *destek vektör makinesi* ve 1 için *karar ağacı*nın verdiğini görürüz. AdaBoost ve destek vektör makinesi ikili sınıflandırmaya odaklanmış görünmektedir ve 3. sınıfta başarımların oldukça düştüğü gözlenmektedir. Bu matrislere dayanarak üç sınıflı sınıflandırma konusunda özellikle rastgele orman, karar ağacı ve çok katmanlı algılayıcının göze çarpan bir başarıya sahip olduğunu söyleyebiliriz.

İlgili sınıflandırma yöntemlerinin önceki bölümde anlatılan ölçütleri Tablo 3'te verilmiştir.

Tablo 3: Yöntemlerin karşılaştırma ölçütleri (30 çalıştırma ortalaması).

Yöntem	Doğruluk	Kesinlik	Geriçağırım	F ölçütü	ROC AUC
AdaBoost	0.8445	0.8254	0.8445	0.8301	0.8610
Çok Katmanlı Algılayıcı	0.8684	0.8699	0.8684	0.8684	0.8848
Destek Vektör Makinesi	0.8492	0.8261	0.8492	0.8229	0.8621
Karar Ağacı	0.8746	0.8765	0.8746	0.8745	0.8877
En Yakın *k* Komşu	0.8629	0.8617	0.8629	0.8600	0.8813
Naïve Bayes	0.8192	0.8045	0.8192	0.8087	0.8417
Rastgele Orman	**0.8793**	**0.8798**	**0.8793**	**0.8787**	**0.8920**

İlgili ölçütlere göre en iyi sonuçlar kalın olarak gösterilmiştir. Tablo 4'ten de görüleceği üzere birinci yaklaşımda en hızlı eğitilen yöntem en yakın *k* komşu olurken, en hızlı tahmin gerçekleştiren yöntem karar ağacı olmuştur.

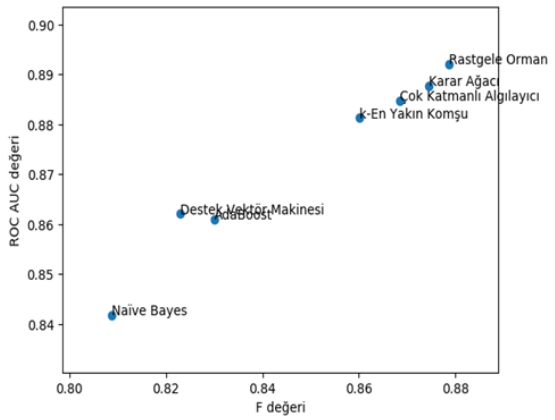
Tablo 4: Yöntemlerin çalışma sürelerinin karşılaştırılması (30 çalıştırma ortalaması).

Yöntem	Eğitim Zamanı	Sınama Zamanı
AdaBoost	0.1062	0.0097
Çok Katmanlı Algılayıcı	0.8838	0.0010
Destek Vektör Makinesi	0.0249	0.0063
Karar Ağacı	0.0037	**0.0002**
En Yakın *k* Komşu	0.0035	0.0055
Naïve Bayes	**0.0034**	0.0003
Rastgele Orman	0.0203	0.0020

Çapraz doğrulama yaklaşımı kullanıldığında (Tablo 5) en hızlı yöntem Naïve Bayes olsa da en iyi sonuçlar çok katmanlı algılayıcı yöntemiyle elde edilmiştir. Yöntemlerin çapraz doğrulama kullanmadan sadece eğitim ve sınama verilerine ayrılarak kullanıldığı ilk yaklaşımda ise rastgele orman en başarılı sonuçları üretmiştir. Rastgele orman yöntemi içerisinde bir topluluk şeklinde kullandığı karar ağacı yönteminin sonuçlarını geliştirmiş ve ondan daha iyi sonuçlar elde etmiştir. Yöntemlerin ROC AUC ve F ölçütü grafiğine baktığımızda (Şekil 7) yöntemlerin bu ölçütlere göre sıralamasını rahatlıkla gözlemleyebiliriz.

Tablo 5: Yöntemlerin çapraz doğrulama sonuçları.

8,5	Çapraz Doğrulama	Zaman
AdaBoost	0.8404 (+/- 0.0002)	1.0288
Çok Katmanlı Algılayıcı	**0.8825** (+/- 0.0004)	11.0373
Destek Vektör Makinesi	0.8625 (+/- 0.0003)	0.4388
Karar Ağacı	0.8788 (+/- 0.0006)	0.0570
En Yakın *k* Komşu	0.8654 (+/- 0.0008)	0.0832
Naïve Bayes	0.8145 (+/- 0.0014)	**0.0455**
Rastgele Orman	0.8795 (+/- 0.0007)	0.2491



Şekil 7: Yöntemlerin çalıştırılması sonucunda elde edilen ROC AUC ve F1 değerleriyle oluşturulan grafik en iyi yöntemin rastgele orman olduğunu gösteriyor.

5 Sonuçlar

Bu makale kapsamında farklı makine öğrenmesi yöntemleri kullanılarak kimlik hırsızı web sitelerinin sınıflandırılması gerçekleştirilmiştir. Bu amaçla hâlihazırda mevcut bir veri kümesi kullanılmıştır. Yöntemler çalışma zamanı ve başarım ölçütleri açısından karşılaştırılmıştır.

Birbirine yakın sonuçlar elde edilmesinin önemli bir nedeni veri kümesinde yer alan bazı özelliklerin, örneklerin ait olduğu sınıfı belirten özellikle güçlü ve orta derecede ilişkili olmasından kaynaklanmaktadır. Buna rağmen yaptığımız karşılaştırma, bir topluluk öğrenme algoritması olan rastgele ormanın, karar ağaçlarının hızlı eğitim ve kestirim özelliklerinden yararlanarak, daha iyi sonuçlar üreten önemli bir model olduğunu göstermiştir. Bu sonuçlara göre rastgele orman yönteminin aşırı uyum sorunu olan karar ağaçlarından bir kümeyi eğiterek, bu soruna uygun bir çözüm bulduğunu söyleyebiliriz. Topluluk öğrenme yöntemlerinde karar ağacının kullanılması da bir bakıma hızlı eğitim ve kestirim özelliklerinden kaynaklanmaktadır.

Gelecekte yapılabilecek uygun bir çalışma varsayılan parametreleri kullanılan bu modellerin parametre analizini yaparak, en uygun parametrelerin bulunması ve özellikle bu çalışmada diğerlerine göre düşük bir başarı gösteren modellerin parametreleri iyileştirildiğinde daha iyi bir başarı gösterip göstermeyeceğini incelemek olacaktır.

6 Kaynaklar

[1] Marshland S. *Machine Learning An Algorithmic Perspective*. 2nd ed. New York, USA, Chapman & Hall/CRC Press, 2015.

[2] Mitchell T. *Machine Learning*. New York, USA, McGraw Hill, 1997.

[3] Alpaydın E. *Yapay Öğrenme*. 3. Baskı. İstanbul, Türkiye, Boğaziçi Üniversitesi Yayınevi, 2017.

[4] Harrington P. *Machine Learning in Action*. New York, USA, Manning Publications, 2012.

[5] Abdelhamid N, Ayesh A, Thabtah F. "Phishing detection based associative classification data mining". *Expert Systems with Applications*, 41(13), 5948-5959, 2014.

[6] Dhamija R, Tygar J D, Hearst M. "Why phishing works?". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montreal, QC, Canada, 22-27 April 2006.

[7] Anti-Phishing Working Group. "APWG Phishing Attack Trends Reports". https://www.antiphishing.org/resources/apwg-reports/, (11.02.2018).

[8] Miyamoto D, Hazeyama H, Kadobayashi Y. "An evaluation of machine learning-based methods for detection of phishing sites". *Australian Journal of Intelligent Information Processing Systems*, 10(2), 54-63, 2008.

[9] Abdelhamid N, Ayesh A, Thabtah F. "Associative classification mining for website phishing classification". *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, USA, 22-25 July 2013.

[10] Aburrous M, Hossain MA, Dahal K, Thabtah F. "Predicting phishing websites using classification mining techniques with experimental case studies". *7th International Conference on Information Technology: New Generations*, Las Vegas, USA, 12-14 April 2010.

[11] Kaytan M. Web Tabanlı Oltalama Saldırılarının Makine Öğrenmesi Yöntemleri İle Tespiti. Yüksek Lisans Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Malatya, Türkiye, 2016.

[12] Kaytan M, Hanbay D. "Effective classification of phishing web pages based on new rules by using extreme learning machines". *Anatolian Journal of Computer Sciences*, 2(1), 15-36, 2017.

[13] Kazemian HB, Ahmed. "Comparisons of machine learning techniques for detecting malicious webpages". *Expert Systems with Applications*, 42(3), 1166-1177, 2015.

[14] Koşan MA, Yıldız O, Karacan H. "Kimlik avı web sitelerinin tespitinde makine öğrenmesi algoritmalarının karşılaştırmalı analizi". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 24(2), 276-282, 2018.

[15] Lakshmi VS, Vijaya MS. "Efficient prediction of phishing websites using supervised learning algorithms". *Procedia Engineering*, 30, 798-805, 2012.

[16] Moghimi M, Varjani AY. "New rule-based phishing detection method". *Expert Systems with Applications*, 53, 231-242, 2016.

[17] Mohammad RM, Thabtah F, McCluskey L. "Intelligent rule-based phishing websites classification". *IET Information Security*, 8(3), 153-160, 2014.

[18] Mohammad RM, Thabtah F, McCluskey L. "Predicting phishing websites based on self-structuring neural network". *Neural Computing and Applications*, 25(2), 443-458, 2014.

[19] Nguyen HH, Nguyen DT. "Machine Learning based phishing web sites detection". *AETA 2015: Recent Advances in Electrical Engineering and Related Sciences. LNEE*, 371, 123-131, 2016.
[20] Sahoo D, Liu C, Hoi SCH. "Malicious URL detection using machine learning: a survey". *ArXiv e-prints, 1701.07179*, 2017.
[21] Haykin S. *Neural Networks and Learning Machines*. 3rd Ed. New Jersey, USA, Pearson Education, 2009.
[22] Zhou ZH. *Ensemble Methods: Foundations and Algorithms*. New York, USA, Chapman and Hall/CRC, 2012.
[23] Freund Y, Schapire RE. "A Decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*, 55(1), 119-139, 1997.
[24] Schapire RE. *Explaining AdaBoost*, Editors: Schölkopf B, Luo Z, Vovk V. Empirical Inference, 37-52, Berlin, Germany, Springer, 2013.
[25] Rumelhart DE, Hinton GE, Williams RJ. "Learning internal representations by back-propagating errors". *Nature*, 323(99), 533–536, 11986.
[26] ZackWeinberg, Wikimedia Commons, File: Svm separating hyperplanes (SVG).svg, https://commons.wikimedia.org/w/index.php?title=File:Svm_separating_hyperplanes_(SVG).svg&oldid=217578095, (11.02.2018).
[27] Hsu CW, Chang CC, Lin CJ. "A Practical Guide to Support Vector Classification". Department of Computer Science, National Taiwan University, Technical Report, https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, (11.02.2018).
[28] Cortes C, Vapnik V. "Support vector networks". *Machine Learning*, 20(3), 1-25, 1995.
[29] Ayhan S, Erdoğmuş Ş. "Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi". *Eskişehir Osmangazi Üniversitesi İBBF Dergisi*, 9(1), 175-198, 2014.
[30] Eviatar Bach, Wikimedia Commons, File: Simple decision tree.svg, https://commons.wikimedia.org/w/index.php?title=File:Simple_decision_tree.svg&oldid=244802879, (11.02.2018).
[31] Onan A. "Şirket iflaslarının tahmin edilmesinde karar ağacı algoritmalarının karşılaştırmalı başarım analizi". *Bilişim Teknolojileri Dergisi*, 8(1), 9-19, 2015.
[32] Cover TM, Hart PE. "Nearest neighbor pattern classification". *IEEE Transactions on Information Theory*, 1967, 13(1), 21-27, 1967.
[33] Breiman L. "Random Forests". *Machine Learning*, 45(1), 5-32, 2001.
[34] UCI Machine Learning Repository, "Website Phishing Data Set", https://archive.ics.uci.edu/ml/datasets/Website+Phishing (11.02.2018)
[35] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, 12, 2825-2830, 2011.
[36] Burman P. "A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods". *Biometrika*, 76(3), 503-514, 1989.