Anatolian Journal of Computer Sciences

Volume:6 No:1 2021 © Anatolian Science

pp:1-10 ISSN:2548-1304

Saldırı Tespit Sistemlerinde Sınıflandırma Yöntemlerinin Kıyaslanması

Cemile İNCE[1], Kenan İNCE[2], Davut HANBAY [3]*

*[*1]İnönü Üniversitesi Bilgi İşlem Daire Başkanlığı, Malatya, Türkiye (cemile.ince@inonu.edu.tr)*

[2]İnönü Üniversitesi Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye (kenanince@gmail.com)

[3]İnönü Üniversitesi Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye (davut.hanbay@inonu.edu.tr)

Received Date : Sep. 9, 2020

Acceptance Date : Nov. 25, 2020

Published Date : Mar. 1, 2021

Özetçe— İnternet hizmetlerinin hayatımızın her aşamasına girdiği günümüzde, kullanılan sistemlerin güvenliği her geçen gün daha da önem kazanmaktadır. Bu anlamda, Saldırı tespit sistemleri (STS) çok önemli yere sahip bir çalışma alanıdır. STS'ler büyük, orta ve küçük ölçekli kuruluşların trafik verilerinin analizinde kullanılır. Bu sistemlerin performansı, hatalı pozitif sınıflandırmalarının az olması ve saldırı türünü doğru kategorize etmesine bağlı olarak değerlendirilir. Birçok farklı yöntem ile STS çalışmaları yapılmakla birlikte, makine öğrenmesi (MÖ) yöntemleri tatmin edici çözümler sunabilmektedir. Bu çalışmada en yaygın kullanılan MÖ tekniklerinden destek vektör makinaları (DVM), rasgele orman (RO), k-NN (k- en yakın komşu), aşırı öğrenme makineleri (AÖM) yöntemleri tanıtılmış ve kıyaslanmıştır. Bu sistemlerin performansını değerlendirmek için veri seti olarak STS'lerin değerlendirilmesinde bir ölçüt olarak kabul edilen NSL-KDD kullanılmıştır. Doğruluk ve F score parametreleri kullanılarak modellerin performansları hesaplanmıştır. En iyi performans AÖM yöntemi ile elde edilmiştir. Hesaplanan doğruluk değeri %99,8, F score değeri %99,9 olarak hesaplanmıştır.

Keywords : *Saldırı tespit sistemi, destek vektör makinaları, aşırı öğrenme, rasgele orman, k-En yakın komşuluk*

1. Giriş

Bilgisayar sistemlerinin son yıllarda çok büyük gelişim göstermesiyle beraber bilgisayar sistemlerine ve servislerine yetkisiz girişler de doğru orantılı olarak artmıştır. Yetkisiz giriş, bilişim teknolojilerinde ve sistem güvenliklerinin sağlanmasında ciddi bir sorundur. Tek bir izinsiz giriş işlemi bilgisayar ve ağ sistemlerinin birkaç saniye içerisinde işlev göremez hale gelmesine sebep olabileceği gibi verilerin çalınması, silinmesi, değiştirilmesi gibi çok büyük sorunları da beraberinde getirebilir. Yetkisiz girişler sadece yazılımsal zarar ile sınırlı kalmayarak donanımsal olarak da sistemleri zarara uğratabilecektir. Bunun gibi birçok nedenden dolayı STS çok önemli çalışma alanlarındandır ve saldırıların önceden tespit edilip önlenmesi gerekmektedir (Wang ve diğerleri, 2017). Geçmişten günümüze birçok STS geliştirilmiştir. Ancak bunların performansı ve gerçek sistemlere uygulanabilirlikleri farklı bir değerlendirme konusudur (Takaoğlu ve diğerleri, 2017).

Bir STS'nin başarısı saldırıyı algılamadaki ve doğru sınıflandırmadaki performansıdır. Hatalı pozitif ve doğru negatif oranını azaltmak için birçok yöntem ve method önerilmiştir. Bu çalışmada DVM, RO, kNN, AÖM yöntemlerini irdeleyip kıyaslamalar gerçekleştirilmiştir. Bu yöntemlerin sınıflandırma problemini ele alma kabiliyetlerinde etkili oldukları kanıtlanmıştır (Gök, 2017). Bu çalışmada gerçekleştirilen uygulamalar standart bir veri seti üzerinde göstermiş oldukları performans değerleri ile doğrulanmıştır. NSL-KDD (Ibrahim ve diğerleri, 2013), KDD'nin geliştirilmiş bir biçimi

olan ve saldırı tespit yöntemlerinin değerlendirilmesinde bir ölçüt olarak kabul edilen bir veri kümesidir.

Literatür incelendiğinde, KDD veri kümesinin varyasyonlarının sıklıkla kullanıldığı ve uygulama yöntemi olarak da MÖ tekniklerinin geçmişten günümüze popüler olduğu görülebilir.

Kaya Ç. ve Yıldız O. (Kaya ve Yıldız, 2014), 2014 yılında gerçekleştirdikleri çalışmada 2007-2013 arasında SCI, SCIE ve EBSCO gibi dizinlerce taranan ulusal ve uluslararası yayınların 65 tanesi incelenmiş, STS'lerde en çok kullanılan MÖ teknikleri karşılaştırılmıştır. Böylece ileride kullanılacak yöntemlere bir öngörü getirilmesi hedeflenmiştir. Yapılan kıyaslama çalışmaları sonucunda en çok kullanılan öğrenme yönteminin YSA, en çok kullanılan veri setinin ise KDD99 veri seti olduğu sonucu çıkarılmıştır.

Özer Ç. Ve Takaoğlu M. (Takaoğlu ve Özer, 2019) , 2019 yılındaki çalışmalarında MÖ tekniklerinden DVM ve naive bayes sınıflandırıcılar kullanılarak bir STS geliştirdiklerini; geliştirdikleri sistemin verilerini çeşitli sistemlerden ve ağ kaynaklarından veri toplayarak sonraki olası güvenlik sorunları için verileri analiz edebilmeyi hedef edinmişlerdir. Sonuç olarak DVM yöntemiyle %71, bayes sınıflandırıcılar ile %79 başarı oranı elde etmişlerdir. DVM'nın naive bayes sınıflandırıcılara nazaran doğruluk oranlarının daha yüksek sonuçlar getirmesi beklenirken daha az başarı oranı elde edilmesini ise; fazla eğitim süresi nedeniyle veri boyutu yüksek olan verilerde çok da verimli çalışamaması olarak belirtmişlerdir.

Kaynar O. Ve ark. (Kaynar ve diğerleri, 2018), 2018 yılındaki çalışmalarında öznitelik seçimleriyle STS gerçekleştirmek konulu çalışmalarında veri seti üzerinde öznitelik seçimi yöntemlerinden kazanım oranı, OneR, ki-kare, bilgi kazancı, gini indeksi gibi birçok veri ön işleme yöntemleri kullanarak bir çalışma gerçekleştirmişlerdir. Sınıflandırma algoritmalarından k-en yakın komşu, DVM, AÖM kullanılmıştır. Öznitelik seçimi algoritmalarının tüm sınıflandırma yöntemlerinde olumlu sonuçlar oluşturduğunu ifade eden çalışmada, k-en yakın komşu sınıflandırma yönteminde en yüksek başarı elde edildiği sonucu belirtilmiştir.

Sağıroğlu Ş. ve ark. (Sa ve diğerleri, 2011), bilgi güvenliği için bir STS geliştirmiş, bu çalışma kapsamında yapay sinir ağları (YSA) ve zeki STS'ler araştırmışlardır. Veri seti olarak KDD'99 veri seti kullanılmış olup en yüksek başarı oranı %97,92 elde edilirken, en düşük başarı oranı %81,93 olarak tespit edilmiştir. Bu çalışma ile literatürde yapılan çalışmalarda çok farklı YSA yapıları ve algoritmalarının kullanıldığı, YSA eğitiminde en çok geriye yayılım algoritmasının kullanıldığı, veri kümesi oluşturmada güncel çalışmaların mevcut olmadığı da yapılan diğer çıkarımlar arasındadır.

Tanrıkulu H. Ve Sazlı M.H. (Tanrıkulu ve Sazlı, 2017), YSA kullanılarak ağ üzerinde akan paketlerin hangi saldırı yönteminin kullanıldığının bulunması amaçlanmıştır. Önce STS'lerin nasıl çalıştığı, ardından saldırı tiplerinden bahsedilmiştir. Makina öğrenmesi yöntemlerinden çok katmanlı algılayıcı (MLP-Multilayer Perceptron) yapay sinir ağı kullanılmış, sonuçlar matlab ortamında değerlendirilmiştir. Yapılan çalışma sonucunda DoS (Denial of Service-Hizmet Reddi) saldırıları motif olarak kullanılmış, tüm saldırıların YSA ya öğretilebildiği sonucuna varılmıştır.

Belavagi M.C. ve Muniyal B. (Belavagi ve Muniyal, 2016), 2016 yılında yapmış oldukları çalışmalarında sınıflandırma modellerinden lojistik regresyon, gauss naive bayes, DVM, RO gibi sınıflandırma yöntemlerini kullanmışlardır. Veri seti olarak NSL-KDD veri seti kullanılmış olup rasgele orman sınıflandırıcıların veri trafiği analizinde daha başarılı sonuçlar oluşturduğu belirtilmiştir. RO yöntemiyle elde edilen başarı oranı %99 olarak tespit edilmiştir.

Tongtong S.U. ve ark. (Tongtong ve diğerleri, 2020) , çalışmalarında NSL-KDD veri seti kullanarak derin öğrenme metoduyla saldırı tespit sistemi gerçekleştirmişlerdir. Bu çalışma ile büyük veriler üzerinde derin öğrenme yöntemlerinin uygulanmasına dayalı bir yöntem geliştirilmiştir. Uygulama sonucu BAT-MC olarak isimlendirdikleri metotla %84.25 doğruluk oranıyla saldırı tespiti gerçekleştirildiğini ifade etmişlerdir.

Kesswani, R. (Choudhary ve Kesswani, 2020), çalışmalarında derin öğrenme yöntemlerini kullanarak CNN ve RNN sinir ağlarını kullanmışlardır. Yine çalışmalarında NSL-KDD veri seti kullanılarak veri seti için %90 doğruluk oranı elde edildiği ifade edilmiştir.

Bu çalışmanın organizasyonu şu şekilde yapılmıştır. Bölüm 2'de materyal ve metot, bölüm 3'te değerlendirme, bölüm 4'te ise sonuç bölümü yer almaktadır.

2. Materyal ve Metot

Bu çalışmanın temel adımlarını veri kümesinin ön işlemden geçirilmesi, sınıflandırma ve sonuç değerlendirmesi aşamalarından oluşmaktadır. Veri ön işleme, sınıflandırıcının performansında ana etmendir. Bu çalışma veri ön işleme adımının performansı birebir etkilediğini de göstermektedir. Bu çalışmanın gerçekleştirilme adımları Şekil 1'de gösterilmiştir.

Bu çalışmanın organizasyonu şu şekilde yapılmıştır. Bölüm 2'de materyal ve metot, bölüm 3'te uygulama, bölüm 4'te ise sonuç bölümü yer almaktadır. Makine öğrenmesi yöntemleri disiplininden geçirilen veri setimiz ileriki çalışmalarımızda derin öğrenme yöntemi uygulanarak sonuçların ileriki çalışmalarda kıyaslaması yapılması planlanmaktadır.

2.1. Kullanılan Veri Kümesi

Deneysel sonuçların gerçekliği ve uygulanabilirliği açısından kullanılan veri kümesi çok önemlidir. Veriler ne kadar gerçekçi olursa sistemin uygulanabilirlik ve kıyaslanabilmeleri o kadar fazla olur. Bu sebeple literatürde sıkça kullanılan NSL-KDD veri kümesi tercih edilmiştir. Bu veri kümesi günümüzde saldırı tespitinde güncel olarak kullanılmaya devam edilmektedir (Ibrahim, L. M. ve diğerleri, 2013)

Şekil 1: İzinsiz giriş tespit sisteminin önerilen modeli.

NSL-KDD veri kümesi 42 nitelikten oluşmaktadır. Bu niteliklerin 4 tanesi kategorik, 6 tanesi binary, 23 tanesi ayrık ve 10 tanesi sürekli veridir. Toplam 4 ana kategori (DoS, U2R, Probe ve R2L) altında 39 saldırı türü içermektedir. Doğal olarak normal kategorisi ile birlikte toplam 5 kategori bulunmaktadır. Bu çalışmada normal ve anormal olmak üzere 2 kategorili sınıflandırma yapılmıştır.

2.2. Veri Ön İşleme

Makine öğrenmesi yöntemlerinin en önemli aşamalarından biri veri ön işleme aşamasıdır. Verilerin ön işleme adımları sonucu olumlu ya da olumsuz etkileyecektir. Veri ön işlemenin ilk adımı verinin sayısallaştırılmasıdır. Sınıflandırıcı, işlenmemiş ham veri kümesini (kategorik, karakter katarı vb. nitelikler içeren) işleyemez. Nümerik olmayan özelliklerin işlemden geçirilerek sayısallaştırılması gerekmektedir. Çünkü sorunsuz ve eksizsiz ve veri kümesi ile işlem yapmak gerçekçi sonuçları doğuracaktır (Aburomman ve Reaz, 2016). Aksi takdirde sınıflandırma sonucu hatalı olacaktır. Veri ön işleme süreci, geliştiriciye ve geliştirme ortamına ek yük anlamına gelir. Numerik olmayan veriler

veri kümesi içerisinden çıkarılır yada veri sayısallaştırma işlemine tabi tutularak düzeltme işlemi gerçekleştirilmiş olur (Dong ve Wang, 2016). Bu niteliklerin çıkarılması sınıflandırma performansında düşüşe sebep olabileceğinden (niteliğin sınıflandırma performansına bağlı olarak) veri ön işleme süreci MÖ yöntemlerinde kaçınılmaz bir adımdır.

2.3. Sınıflandırma

Sınıflandırma aşamasının amacı özetle, verinin niteliklerine göre etiketlenmesidir. Kullanılan veri kümesinin 42. sütunu olan çıkış verilerimizin normal ya da anormal şeklinde sınıflandırma yapacağımız yani etiketleme yapacağımız veriler mevcuttur.

STS'nin temel işlevi, saldırı ya da değil şeklinde bir analiz yapmaktır. Literatürde analiz hesabı yapan birçok sınıflandırma algoritması mevcuttur. YSA, DVM, RO, k-en yakın komşu ve bayes sınıflandırıcılar gibi birçok sınıflandırma yöntemi ve algoritmaları mevcuttur. Bu çalışmada seçilen veri kümesine en uygun ve en etkili sınıflandırma yöntemlerinden DVM, RO ve AÖM yöntemleri ele alınmıştır (Taşçı E. ve diğerleri, 2017).

2.3.1. Destek Vektör Makineleri

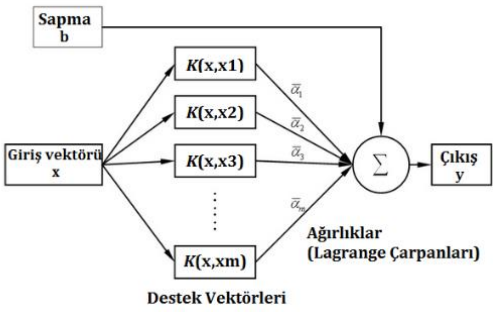
DVM yöntemi, sınıflandırma ve regresyon problemlerini çözmek için 1995 yılında Vapnik tarafından önerilen sınıflandırma yöntemidir (Cortes ve Vapnik, 1995). DVM, sınıflandırma yapmak için farklı veri ve kategorilerde farklı disiplinlerden eğitilmiş denetimli bir öğrenme tekniğidir (Jha ve Ragha, 2013). DVM sınıflandırma yönteminde sınıflandırıcı en iyi sınıflandırmayı yapabilmek için bir veya birden çok hiperdüzlem oluşturur (Hofmeyr et al., 1998). Bu hiper düzlemler lineer ya da lineer olmayan veri sınıflandırılmasında kullanılabilir. Bu hiper düzlemleri oluşturmak için çekirdek fonksiyonları kullanılır. Doğrusal, polinomsal, radyal, sigmoid ve temel çekirdek fonksiyonları gibi çekirdek fonksiyonları mevcuttur (Abdalla ve Erdoğmuş, 2014). Son yıllarda görüntü işleme ve örüntü tanıma alanlarındaki yaygın kullanımı nedeniyle DVM sınıflandırıcılar konusunda çok fazla umut verici araştırmalar geliştirilmiştir. Şekil 2, DVM mimarisini göstermektedir. DVM sınıflandırıcı uygularken radyal tabanlı fonksiyon ya da gauss diye adlandırdığımız çekirdek fonksiyonu kullanılmıştır. DVM uygulamalarında çekirdek fonksiyonu ve parametrelerinin belirlenmesi önemlidir. Çünkü bu fonksiyon dönüştürülmüş nitelik uzayını tanımlar (Kotsiantis, 2007).

Bu çalışmada kullanılan kernel çekirdek fonksiyonunun matematiksel formülü ise şu Denklem 1,2 ve 3'te ifade edilmiştir.

$$\varphi\,(xi, xj) \;=\; exp\,\{-\gamma\,||xi - xj\,||\} \tag{1}$$

$$K(x,y) = e\text{^}-\gamma^{|x-y|\text{^}2} \tag{2}$$

$$Min\; w, b, \xi \;\; \frac{1}{2} w^T * W + C \sum_{i=1}^{n} \xi \tag{3}$$



Şekil 2: DVM mimarisi

DVM'nin avantajı, minimum parametre ayarının gerekli olmasıdır. Dezavantajları ise, eğitim setinin her bir örneği için bir Gauss fonksiyonunun gerekliliklerini içermesi, böylece sınıflandırmada olduğu gibi binlerce örneğe sahip çok büyük veri kümelerinde eğitim süresinin artmasına ve performans düşmesine sebep olabilmesidir. ε_i $(i = 1,2, \dots, n)$ Kısıtlamalar da aşağıdaki gibi ifade edilir.

$$(w.x_i - b) \geq +1 - \varepsilon_i \, , y_i = +1 \tag{4}$$

$$(w.x_i - b) \geq -1 + \varepsilon_i \, , y_i = -1 \tag{5}$$

DVM sınıflandırıcıda kullanılan lagrange çarpanları hesabı ise şu şekilde hesaplanmaktadır:

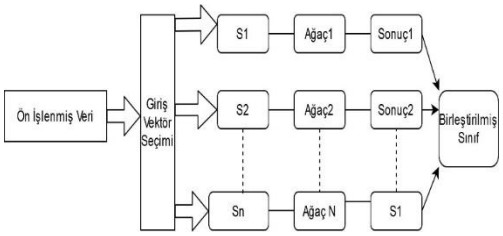
$$L_p = \frac{1}{2}\|w^2\| + (C\sum_{i=1}^{n}\varepsilon_i - \sum_i \alpha_i \{y_i(x_i.w - b) - 1 + \varepsilon_i\} - \sum_i \mu_i \varepsilon_i) \tag{6}$$

Lagrange çarpanları denklem (6)'da gösterildiği gibi matematiksel olarak ifade edilir.

2.3.2. Rasgele Orman

RO, yetkisiz girişleri algılama tespitinde sınıflandırma ve regresyon analizi için kullanılan topluluk sınıflandırmasıdır. RO'lar, eğitim aşamasında çeşitli karar ağaçları ve çoğunluğa göre etiketler oluşturarak çalışırlar. RO'larin karar ağacı algoritmalarından farkı temel olarak kök düğümü bulma ve düğümleri bölme işlemlerinin rasgele çalışıyor olmasıdır. Bu çalışmada RO yönteminin de ele alınmasının sebebi, gürültü ve aykırı değer saptanmasında iyi olmaları, aşırı öğrenme (overfittting) zorluklarının olmamasıdır. Ayrıca veri seti özellikleri arasından en önemli özelliği tanımlamak için en uygun yöntemlerden birisidir. Böylece özellik çıkarımı en doğru şekilde uygulanarak başarı oranının en yüksek oranlara çıkabilmesi sağlanmış olur.

Şekil 3'de RO algoritmasının çalışma sistemi şematik olarak gösterilmiştir.

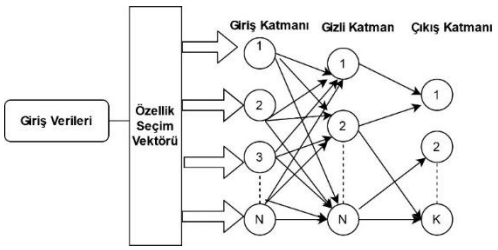


Şekil 3: RO Mimarisi

RO, bir dizi özellik alt kümesi kullanarak n farklı ağaç oluşturur. Her ağaç bir sınıflandırma sonucu üretir ve sınıflandırma modelinin sonucu oy çokluğuna bağlıdır. Örnek, en yüksek oyu alan sınıfa verilir. Daha önce elde edilen sınıflandırma sonuçları, RO'nin bu tür verilerin sınıflandırılmasında makul olarak uygun olduğunu göstermektedir.

2.3.3. Aşırı Öğrenme Makineleri

AÖM, tek veya çoklu gizli katman içeren ileri beslemeli yapay snir ağlarıdır denilebilir (Huang ve diğerleri, 2004). AÖM, çeşitli sınıflandırma, kümeleme, regresyon ve özellik problemlerini çözmek için kullanılabilir. AÖM algoritmasının çalışma sistemi Şekil 4'te ifade edilmiştir.



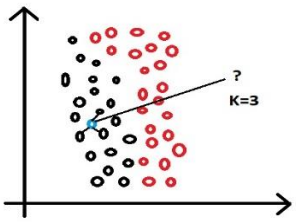
Şekil 4: ELM Mimarisi

Bu öğrenme algoritması giriş katmanını, bir veya daha fazla gizli katmanı ve çıkış katmanını içerir. Geleneksel sinir ağlarında, girdi ve gizli katman ağırlıklarının ayarlanması hesaplama açısından pahalı ve zaman alıcıdır. Çünkü sistem eğitilene kadar iterasyonlar yenilenecektir. Temel olarak YSA ile aynı mimariye sahip olmasına rağmen AÖM'lerde giriş katmanı ve eşik değerleri rasgele atanmaktadır. Çıkış değerleri de buna bağlı olarak hesaplanmaktadır (Faruk Ertuğrul ve Kaya, 2014). YSA'nın öğrenebilmesi için izlediği yöntem ise ağırlıklar, eşik değerleri, transfer fonksiyonları leri ve geri yayılım algoritmaları ile hata değerleri hesaplanarak güncelleme ve en iyileme temelin mantığına dayanmaktadır (Ertuğrul, 2016). AÖM, oldukça karmaşık veri kümelerinde daha iyi performans gösterme yeteneğine sahiptir.

2.3.4. En Yakın Komşuluk

Sınıflandırma yöntemlerinden biri olan k-NN algoritması k değerine bakarak yakınındaki en yakın komşulara olan uzaklıkların hesaplanmasına dayalı çalışan bir algoritmadır(Kaytan ve Hanbay, 2017). K-NN algoritmasının en büyük avantajı uygulama kolaylığıdır. Ancak zaman karmaşıklığı ve bellek gereksinimi algoritmanın negatif taraflarıdır(Amal ve Ahmed, 2011).

K-NN algoritmasında komşuluklar arası uzaklık hesaplaması yapılırken çeşitli algoritmalar kullanılır. Bunlardan en yaygınları öklid, manhattan, minkowski ve chebyshev uzaklık algoritmalarıdır (Wang ve diğerleri,2017). En yakın komşu algoritması çoğu sınıflandırma algoritmalarına göre daha hızlı çalışır (Taşcı ve Onan, 2016). Şekil 5'te üç komşuluk üzerinden k-nn algoritması şematize edilmiştir. Şekilden de anlaşılabileceği gibi, K=3 alındığında en yakın 3 komşuluk mesafeleri hesaplanarak işlem gerçekleştirilir.



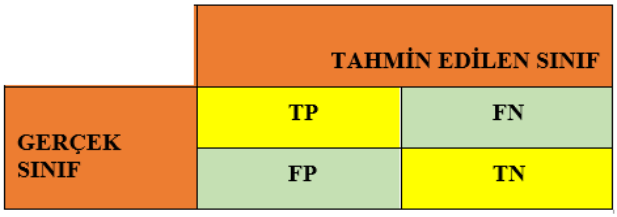
Şekil 5: k-NN Algoritması Şeması

3. Değerlendirme

NSL-KDD veri seti 65535 satırlı, 42 öznitelikli büyük veri kümesi örneklerindendir. Uygulama aşamasından önce veriler önişleme aşamasından geçirilip nümerik olmayan veriler çıkartılmıştır. Böylece sonuçların daha tutarlı ve tekrarlanabilir sonuçlar oluşturması sağlanmıştır.

Kullanılan verinin %5, %10 ve tamamı kullanılarak farklı testler gerçekleştirilmiştir. Verinin %5'i için 20 kayıttan biri, %10'u için her 10 kayıttan biri düzenli olarak alınarak bu işlem gerçekleştirilmiştir. Yapılan çalışmada verinin bu şekilde işlenmesinin sebebi, verinin düzgün dağılımlı olup olmadığının kontrol edilmesidir.

Gerçekleştirilen çalışmada performans değerlendirilirken karışıklık (confusion) matris kullanılmıştır. Sınıflandırma algoritmaları performans değerlendirmesi yaparken sadece doğruluk oranına bakılması çalışma sonuçlarının objektif değerlendirilmesini engelleyecektir. Doğruluk oranına ek olarak duyarlılık (recall), kesinlik (precision) değerlerine de bakılması büyük önem arz etmektedir. Karışıklık matrisinin sonuç alanları Şekil 6'da gösterilmiştir. Karışıklık matrisinin değerleri kullanılarak doğruluk (accuracy), duyarlılık (recall) ve kesinlik (precision) performans ölçütleri hesaplanabilir.



Şekil 6: Karışıklık Matrisin şematize edilmesi

Doğruluk değeri modelde doğru tahmin ettiğimiz alanların toplam veri kümesine oranı ile hesaplanmaktır.

$$Doğruluk = \frac{TP + TN}{TP + FN + FP + TN}$$

Duyarlılık değeri pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren metrik değere verilen genel addır.

$$Duyarlılık = \frac{TP}{TP + FN}$$

Kesinlik değeri, pozitif olarak tahmin ettiğimiz değerlerin gerçekte kaç tanesinin pozitif olduğunu test etmemize yarayan metrik değerin genel adıdır.

$$Kesinlik = \frac{TP}{TP + FP}$$

Bu metrik değerler hesaplandıktan sonra F1 score değerimiz hesaplanmalıdır. Doğruluk değeri yerine F1 score kullanmamızın nedeni eşit dağılmayan veri kümelerinde, ya da dağılımı bilinmeyen veri setlerinde tüm hata maliyetlerini içeren F1 score daha gerçekçi sonuçları gözler önüne sermektedir.

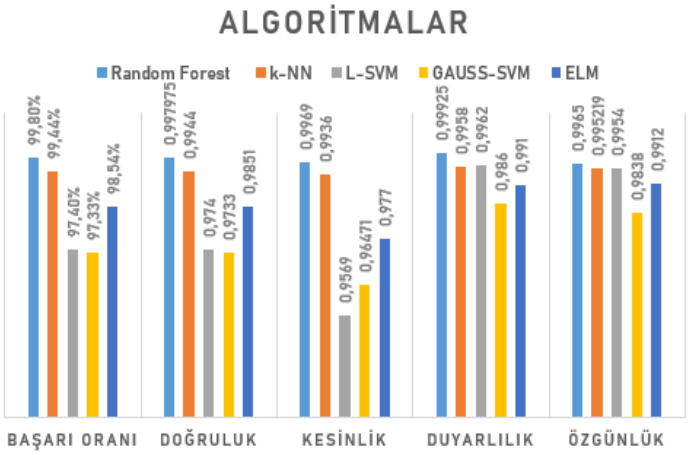
$$F1 = 2 * \frac{kesinlik * duyarlılık}{kesinlik + duyarlılık}$$

Özgünlük değeri, yoku tahmin etme etkinliği olarak yorumlanabilir. F score değeri hesaplanırken özgünlük değeri kullanılmaz.

3.1. Uygulama Sonuçları

Çalışmada kullanılan algoritmalar kıyaslandığı zaman en yüksek performanstan düşüğe doğru bütün kategorilerde (doğruluk, kesinlik, duyarlık, özgünlük) RO, kNN, AÖM ve DVM şeklinde sıralandığı gözlemlenmiştir. RO %99.9, kNN %99.7, AÖM %98, L-DVM %97.5, Gaus DVM %97.4 başarı oranlarına sahip sınıflandırmalar gerçekleştirilmiştir.

Şekil 8 incelendiğinde F score oranının doğruluk oranı ile orantılı olduğu görülmektedir. Bu da verimizin eşit dağılımlı bir veri olduğunu göstermektedir. Şekil 8'de gösterilmiş olan F ölçütü incelendiğinde RO sınıflandırma yönteminin neredeyse verilerin tamamında en uygun yöntem olduğu görülmektedir. Saldırı tespitinde kullanılacak en uygun sınıflandırma yönteminin sırasıyla RO, kNN, ELM ve DVM şeklinde olduğu; büyük veri setlerinde RO yönteminin diğer yöntemlerden daha uygun olduğu görülmüştür.



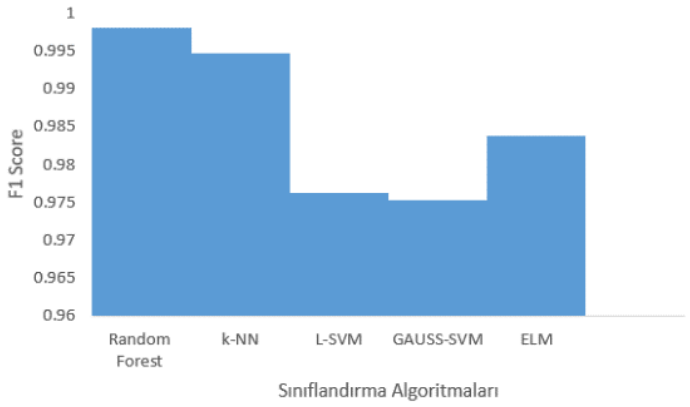
Şekil 7: Algoritmaların kıyaslanma değerleri

Random Forest: Rasgele Ağaçlar

SVM: Destek Vektör Makinaları (DVM)

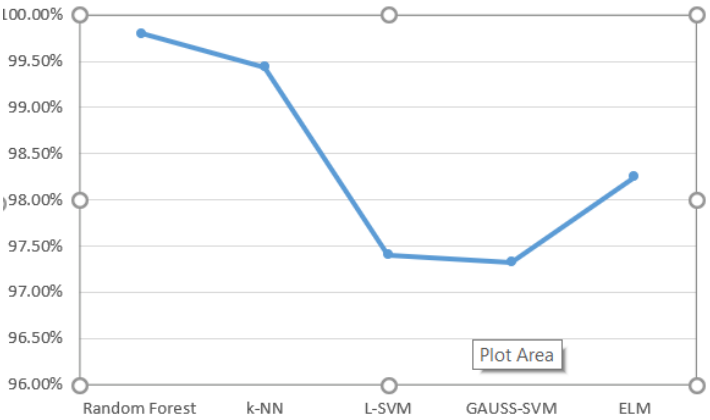
ELM: Aşırı Öğrenme Makineleri

k-NN: k En Yakın Komşuluk



Şekil 8: Sınıflandırma Algoritmalarının F Score Oranları

Şekil 9'de başarı yüzdeleri hesaplanmış olup, DVM haricinde diğer yöntemlerin %100 e yakın bir F ölçütlük oranıyla saldırı tespit işlemini gerçekleştirilebildiği görülmüştür. DVM, çok güçlü bir sınıflandırma yöntemi olmasına karşın diğer yöntemlerden daha az başarı göstermesi; büyük boyutlu verilerde performans düşüklüğünün görülebileceği gerçeğini gözler önüne sermiştir.



Şekil 9: Sınıflandırma Algoritmaları Başarı yüzdeleri kıyaslanması

4. Sonuçlar

Sınıflandırma yöntemlerinden RO, DVM, k en yakın komşu algoritması, AÖM yöntemleri kullanılarak sınıflandırma başarı oranları kıyaslanmıştır. Yapılan kıyaslamalar neticesinde en yüksek başarı oranı RO algoritması ile %99.9 başarım oranı elde edilmiştir. Ayrıca seçmiş olduğumuz veri setinin ön işlemeden geçirildikten sonra düzenli dağılım gösteren veri seti olduğu görülmüştür. Ayrıca birçok uygulamada çok başarılı bir sınıflandırma yöntemi olan DVM'nın kullanılan iki varyasyonu da bu çalışmada çok düşük başarım göstermiştir. Bu durum, uygulama alanına göre algoritma seçiminin ne denli önemli olduğunu göstermiştir.

Klasik makine öğrenmesi yöntemleri çokça irdelenip araştırma alanı bulmuştur. Günümüzde derin öğrenme yöntemleri gibi popüler yöntemler de mevcuttur. Derin öğrenme yöntemlerinin klasik makine öğrenmesi yöntemlerine göre çok daha fazla donanıma bağlılık dezavantajları bulunmaktadır. Ayrıca saldırı tespit sistemlerinde derin öğrenme yöntemleri ile yakalanan başarı oranları klasik makine öğrenmesi yöntemleri kadar başarılı olamamıştır.

Teşekkür

Bu çalışmanın gerçekleştirilmesinde, FBG-2018-1107 ve FBG-2020-2143 numaralı projeler kapsamında, vermiş oldukları maddi ve manevi destekten dolayı, İnönü Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimine teşekkür ederiz.

5. Kaynakça

Abdalla, S., & Erdoğmuş, Ş. (2014). Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *Eskişehir Osmangazi Üniversitesi Iktisadi Idari Bilimler Fakültesi Dergisi*, *9*, 175–198.

Aburomman, A. A., & Reaz], M. B. [Ibne. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372. https://doi.org/https://doi.org/10.1016/j.asoc.2015.10.011

Amal, M.-A., & Ahmed, B.-A. (2011). Survey of Nearest Neighbor Condensing Techniques. *International Journal of Advanced Computer Science and Applications*, *2*(11). https://doi.org/10.14569/ijacsa.2011.021110

Belavagi, M. C., & Muniyal, B. (2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Procedia - Procedia Computer Science*, *89*, 117–123. https://doi.org/10.1016/j.procs.2016.06.016

Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. *Procedia Computer Science*, *167*(2019), 1561–1573.

https://doi.org/10.1016/j.procs.2020.03.367

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411

Dong, B., & Wang, X. (2016). Comparison deep learning method to traditional methods using for network intrusion detection. *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 581–585. https://doi.org/10.1109/ICCSN.2016.7586590

Ertuğrul, Ö. F. (2016). Aşırı Öğrenme Makineleri ile biyolojik sinyallerin gizli kaynaklarına ayrıştırılması. *DÜMF Mühendislik Dergisi*, *7*(1), 41–50.

Faruk Ertuğrul, Ö., & Kaya, Y. (2014). A detailed analysis on extreme learning machine and novel approaches based on ELM. *American Journal of Computer Science and Engineering*, *1*(5), 43–50. http://www.openscienceonline.com/journal/ajcse

GÖK, M. (2017). MAKİNEÖğrenmesiYöntemleriİleAkademik BaşarininTahmiEdilmesi. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, *5*(3), 139–148. https://dergipark.org.tr/gujsc/issue/31140/311082

Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion Detection Using Sequences of System Calls. *J. Comput. Secur.*, *6*(3), 151–180.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, *2*, 985–990 vol.2. https://doi.org/10.1109/IJCNN.2004.1380068

Ibrahim, L. M., Taha, D. B., & Mahmod, M. S. (2013). A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network. *Journal of Engineering Science and Technology*, *8*(1), 107–119.

Jha, J., & Ragha, L. (2013). Intrusion Detection System using Support Vector Machine. *IJAIS Proceedings on International Conference and Workshop on Advanced Computing 2013*, *ICWAC*(3), 25–30.

KAYA, Ç., & YILDIZ, O. (2014). Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz. *Marmara University Journal of Science*, *26*(3), 108. https://doi.org/10.7240/mufbed.24684

KAYNAR, O., ARSLAN, H., GÖRMEZ, Y., & IŞIK, Y. E. (2018). Makine Öğrenmesi ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti. *Bilişim Teknolojileri Dergisi*, 175–185. https://doi.org/10.17671/gazibtd.368583

Kaytan, M., & Hanbay, D. (2017). Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines. *Anatolian Journal of Computer Sciences*, *2*(1), 15–36. https://dergipark.org.tr/download/article-file/333655

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica (Ljubljana)*, *31*.

Sa, Ş., Lu, Ğ. I. R. O. Ğ., Yolaçan, E. N., & Lu, U. Y. Ğ. (2011). ZEKİ SALDIRI TESPİT SİSTEMİ TASARIMI ve GERÇEKLEŞTİRİLMESİ. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, *26*(2), 0. https://doi.org/10.17341/gummfd.74383

TAKAOĞLU, M., & ÖZER, Ç. (2019). Saldırı Tespit Sistemlerine Makine Öğrenme Etkisi. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, *3*(1), 11–22. https://doi.org/10.33461/uybisbbd.558192

Tanrıkulu, H., & Sazlı, M. H. (2017). *Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması*. *January 2008*.

Taşcı, E., & Onan, A. (2016). K- En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi. *Xviii.AkademiBilişim Konferansi*, 8.

Wang, H., Gu, J., & Wang, S. (2017). An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems*, *136*, 130–139. https://doi.org/https://doi.org/10.1016/j.knosys.2017.09.014

Tongtong, S.,Sun, H., Wang, S.,Li, Y.(2020). *BAT:Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset*

Choudhary, S., & Kesswani, N. (2020) *Analysis of KDD- Cup'99, NSL-KDD and UNSW-NB15 Datasets Using Deep Learning in IoT* Peer-review under responsibility of the scientific committee of the International Conference on Computational Intelligence and Data Science 10.1016/j.procs.2020.03.367