Academic Platform Journal of Engineering and Science

journal homepage: http://apjes.com/

Anormal Trafik Tespiti için Veri Madenciliği Algoritmalarının Performans Analizi

*[1] Ünal Çavuşoğlu, [2] Sezgin Kaçar
[1]Sakarya Üniversitesi, Bilgisayar ve Bilişim Fakültesi, Bilgisayar Mühendisliği, Sakarya, unalc@sakarya.edu.tr,
[2] Sakarya Üniversitesi, Teknoloji Fakültesi, Elektrik Elektronik Mühendisliği, Sakarya, skacar@sakarya.edu.tr,

Araştırma Makalesi Geliş Tarihi: 25.04.2018 Kabul Tarihi: 16.11.2018

Öz

Bu çalışmada, bilgisayar ağ trafiğinde tehlike oluşturabilecek zararlı trafiğin tespit edilmesi için kullanılan veri madenciliği algoritmalarının performans değerlendirilmesi gerçekleştirilmiştir. İlk olarak, farklı özellik çıkarım algoritmaları ile NSL-KDD veri setinden nitelik çıkarım işlemi gerçekleştirilmiştir. Bu işlem sonucunda farklı niteliklerden oluşan yeni veri setleri oluşturulmuştur. Bu veri setleri üzerinde farklı veri madenciliği algoritmaları kullanılarak anormal trafik tespiti için testler yapılmıştır. Yapılan testler sonucunda, farklı veri madenciliği ve özellik çıkarım algoritmalarının performans değerlendirmesi sunulmuştur.

Anahtar Kelimeler: Anormal trafik tespit sistemi, veri madenciliği, performans analizi, özellik çıkarım algoritmaları

The Performance Analysis of Data Mining Algorithms for Anomaly Detection

*[1] Ünal Çavuşoğlu, [2] Sezgin Kaçar
[1] Sakarya University, Computer and Informatics Faculty, Computer Engineering Department, Sakarya
[2] Sakarya University, Faculty of Technology, Department of Electrical and Electronics Engineering, Sakarya
unalc@sakarya.edu.tr
skacar@sakarya.edu.tr

Abstract

In this study, performance evaluation of data mining algorithms used to detect harmful computer network traffic is realized. Firstly, feature selection process is performed from the NSL-KDD dataset with different feature selection algorithms. As a result of this process, different datasets are created by combining different attributes. Performans tests are conducted for the detection of anormal traffic using different data mining algorithms on these data sets. As a result of the tests, performance evaluation of different data mining and feature selection algorithms is presented.

Keywords: Anomaly traffic detection, data mining, performance analysis, feature selection algorithms

1. GİRİŞ

Bilişim ve iletişim teknolojilerinde meydana gelen hızlı gelişmeler ile birlikte, bu sistemlere gerçekleştirilen saldırılar ciddi oranlarda artış göstermektedir. Elektronik sistemlere yapılan saldırıların önlenmesi ve veri güvenliğinin korunması, bireyler ve kurumların en önemli ihtiyaçlarından birisi haline gelmiştir.

Ayrıca günümüzde özellikle internetin yaygınlaşması ile veri boyutları oldukça büyümüştür. Sistemlerin ve veri güvenliğinin sağlanması için birçok farklı yaklaşım ve yöntem üzerinde çalışmalar devam etmektedir. Veri güvenliğinin sağlanması, verinin erişim iznine sahip olmayan kişilerin veriye erişiminin önlenmesi, veriyi ele geçirmesi ve veri üzerinde değişim gerçekleştirmesi veya veriyi bozmalarının engellenmesi gibi işlemlerin sağlanması ile mümkün olmaktadır [1-3].

*Sorumlu Yazar: Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, unalc@sakarya.edu.tr

Doi: 10.21541/apjes.418519

Anormal içeriğin tespit edilmesi için tasarlanan saldırı tespit sistemlerinde kullanılan yapılar istatistiki metotlar, bilgi tabanlı uzman sitemler ve makine öğrenmesi teknikleri olmak üzere genel olarak üç kategoriye ayrılmaktadır. Saldırıların fark edilmesi ve engellenmesinde makine öğrenmesi teknikleri saldırı tespit sistemi tasarımlarında yaygın olarak kullanılmaktadır. Bu sistemler sayesinde ağ trafiği üzerinde normal ve anormal yani saldırı olarak etiketlenecek içeriğe sahip veri fark edilerek, sistemin saldırılardan zarar görmesi engellenmeye çalışılmaktadır. Ağ trafiği üzerinde inceleme yapan saldırı tespit sistemleri, iki yaklaşım ile çalışmaktadır. Anormallik tespit sistemleri sistemin normal trafiği ile uyumunu inceleyerek, anormal durumu tespit etmeye çalışmaktadır. Yanlış kullanım tespit sistemleri ise, sistemde kayıtlı olan yanlış kullanım imzaları/ kuralları üzerinden işlem gerçekleştirmektedirler. Fakat bu sistemler yeni saldırılara karşı oldukça savunmasız kalmaktadırlar[4-7].

Makine öğrenmesi teknikleri içinde yer alan veri madenciliği algoritmaları da saldırı tespit sistemlerinin tasarımlarında kullanılan en önemli bileşenlerinden birisidir. Veri madenciliği algoritmaları, büyük veri setlerinde depolanan veriyi kullanarak, veri üzerinden anlamlı ve işe yarayacak bilginin çıkarımını sağlamaktadır. Saldırı tespit sistemleri tasarımında da ağ trafiğinde elde edilen veri kullanılarak sistem eğitilmekte ve anormal durumların tespiti için bir tasarım gerçekleştirilmektedir. Veri madenciliği algoritmaları genel olarak sınıflandırma, kümeleme ve birliktelik analizi gibi yaklaşımlarla veri üzerinde işlem gerçekleştirmektedir.

Lee ve arkadaşları bilgisayar ağları üzerinde veri madenciliği yöntemleri kullanarak normal ve anormal trafiğin ayırt edilmesi için bir çalışma gerçekleştirmişlerdir. Saldırı tespiti için farklı algoritmaları kullanan uygulamalar geliştirmiş ve sonuçlarını değerlendirmişlerdir[8]. Amor ve arkadaşları, Naive Bayes algoritması ile yeni bir saldırı tespit sistemi tasarlamış ve KDD-99 veri seti üzerinde yapmış olduğu uygulama sonuçlarını karar ağacı algoritmasının sonuçları ile karşılaştırmışladır[9].

Blowers ve Williams çalışmalarında, makine öğrenme algoritmalarının siber güvenlik operasyonlarında kullanımını detaylı bir şekilde incelemiş ve makine öğrenme tabanlı saldırı tespit sistemleri analiz edilmiştir[10]. Nadiammai ve Hemalatha çalışmalarında anormal davranışların tanımlı olduğu bir saldırı tespit sistemi veri tabanı oluşturmuş ve 4 farklı aşamada işlem gerçekleştiren bir sistem önermişlerdir. KDD-99 veri seti üzerinde testler gerçekleştirmiş ve yüksek başarım oranında tespit gerçekleştirildiğini göstermişlerdir[11].

Patel ver Panchal çalışmalarında, kural tabanlı saldırı tespit sistemi SNORT ve paket başlık ve trafik anormallik tespit sistemlerinin birlikte kullanıldığı hibrit bir saldırı tespit sistemi sunmuşlardır. Sistem tasarımında veri madenciliği algoritması olarak k-ortalama ve sınıflama regresyon ağacı yöntemleri kullanılmıştır. Önerilen sisteme ait testler KDD-99 veri seti üzerinde yapılmıştır[12]. Li ve arkadaşları veri seti üzerinde kritik öneme sahip niteliklerin belirlenmesini sağlayarak, özellik çıkarımı işlemi gerçekleştirmişlerdir. Önerilen sistemde karar destek makineleri ve karınca koloni algoritmaları kullanılmıştır[13]. Horng ve arkadaşları hiyerarşik kümeleme ve destek vektör makinesinin kullanıldığı bir saldırı tespit sistemi sunmuşlardır. Makalede özellik çıkarım metotlarıda kullanılarak başarımın artırılması hedeflenmiştir[14]. Lin ve arkadaşları KNN(K en yakın komşu) sınıflama algoritması ile bir saldırı tespit sistemi önermişlerdir. Sistemin testlerini KDD-99 veri seti üzerinde farklı saldırı tipleri için yapmışladır[15]. Jiang ve arkadaşları birliktelik kural çıkarımı yaklaşımı ile log kayıtları üzerinden işlemlerini gerçekleştiren bir saldırı tespit sistemi tasarlamışlardır. Çalışmada önerilen sistemin başarılı şekilde atakları tespit ederek, saldırıları engellediği vurgulanmıştır. [16].

Kumar ve Bhaskar farklı kümeleme tekniklerinin kullanıldığı bilgisayar ağları üzerinde anormal durumların tanımlanması için bir sistem tasarlamışlardır. Çalışmada veri seti olarak web kayıtları kullanılmıştır [17]. Sharma ve arkadaşları karar ağaçları ve genetik algoritmaların birlikte kullanıldığı bir saldırı tespit sistemi önermişledir. Önerilen sistem KDD-99 veri seti üzerinde test edilmiştir ve DoS saldırılarında yüksek başarı oranında tespit yaptığı gösterilmiştir[18].

Bu çalışmada, bilgisayar ağlarında anormal trafiğin tespiti için kullanılan farklı veri madenciliği algoritmalarının performans değerlendirmesi yapılmıştır. Çalışmada farklı özellik çıkarım yöntemleri kullanılarak NSL-KDD veri üzerinden, daha az nitelik sayısına sahip farklı veri setleri oluşturulmuştur. Bu veri setleri üzerinde farklı veri madenciliği algoritmalarının anormal trafik tespitindeki performans analizleri gerçekleştirilmiştir. Performans test sonuçları karşılaştırılarak, yüksek oranda doğruluk ve kısa zamanda tespit gerçekleştiren algoritmalar belirlenmiştir.

Belirlenen veri setleri ve algoritmalarının yüksek performanslı bir anormal trafik tespit sistemi oluşturmak için kullanılabileceği ifade edilmiştir. Literatürdeki çalışmalardan farklı olarak, farklı özellik çıkarım tekniklerinin, farklı veri madenciliği algoritmaları ile uyumu test edilmiş ve test edilen yöntemler içinden yüksek başarıma sahip olanlar belirlenmiştir. Bu makalenin amacı, işlem yükünün azaltılması ve performansın artırılması için, daha az niteliğe sahip veri setleri oluşturmak ve bu veri setleri üzerinde yüksek oranda başarıma sahip veri madenciliği algoritmalarının tespit edilmesidir.

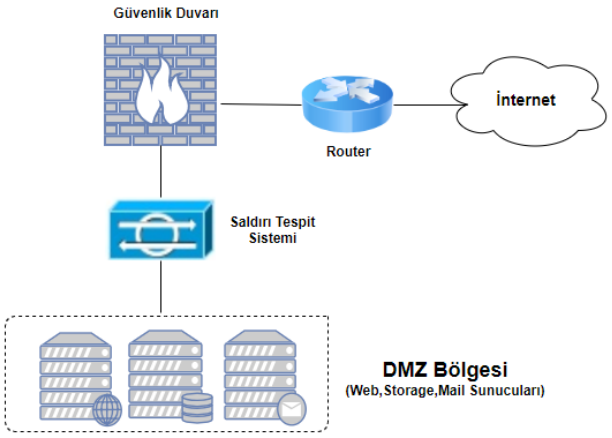
Makalede giriş bölümünün ardından, ikinci bölümde saldırı tespit sistemleri, NSL-KDD veri seti ve Weka programı hakkında bilgi verilmiştir. Üçüncü bölümde, test işlemlerinde kullanılan veri madenciliği algoritmaları ve özellik çıkarım teknikleri açıklanmıştır. Sonrasında performans testlerinin nasıl yapıldığı açıklanmış ve

performans testleri gerçekleştirilmiştir. Son bölümde çalışma ile ilgili sonuç ve değerlendirmelere yer verilmiştir.

2. TEORİK ALTYAPI

2.1. Anormal Trafik Tespit Sistemleri

Anormal trafik tespit sistemleri bilgisayar ağları üzerindeki trafiği analiz ederek, zararlı içeriğe sahip olan trafiğin tespit edilmesini sağlamak için kullanılmaktadır. Saldırı tespit sistemleri, saldırı algılama yöntemine göre iki farklı kategoride incelenebilirler. İmza tabanlı sistemler, sistemde kayıtlı olan bilgiye dayanarak gelen veriyi inceler ve tespit gerçekleştirir[19,20]. Anormal davranış tespit sistemlerinde ise sistem üzerinde tanımlanmış, normal ve anormal içeriğe ait örnekler üzerinden geliştirilen bir model ile trafik üzerinde bir tespit gerçekleştirilmektedir. Şekil 1'de saldırı tespit sistemlerine ait genel bir tasarım görülmektedir.



Şekil 1. Anormallik tesit sistemi genel yapısı

Şekil 2'de veri madenciliği algoritmaları kullanılarak tasarlanabilecek olan bir saldırı tespit sistemine ait genel blok diyagram verilmiştir. Bu sistemde örnek bir veri seti trafiği sisteme verilerek, eğitim ve test veri setleri ayrılmaktadır. Eğitim veri seti üzerinde sistemin trafik içeriğini tanıması sağlanmakta ve daha önce sistem tarafından görülmemiş test veri seti ile de sistemin başarımı test edilmektedir. Sisteme verilen veri seti üzerinde kullanılacak olan veriyi daha iyi hale getirmek, veri boyutunu azaltmak için özellik çıkarım algoritmaları kullanılmaktadır.

2.2. NSL-KDD Veri Seti

KDD-CUP99 veri seti [21] bilgisayar ağ trafiğinde anormalliklerin tespit etmek için geliştirilen sistemlerin test edilmesi için yaygın olarak kullanılan bir veri kümesidir. Bu veri seti farklı saldırı tipi ve türlerinde birçok kayıt içermektedir. Fakat bu veri seti üzerinde gerçekleştirilen çalışmalarda veri setinde test edilen sistemlerin başarımını olumsuz etkileyen bazı durumların olduğu tespit edilmiştir. Bu problemin çözümü için, KDD CUP99 veri setindeki bazı kayıtlar elenerek, önerilen sistemlerin testi için NSL-KDD [22] adı verilen yeni bir veri seti kullanımı önerilmiştir. Bu sayede veri setinin daha küçük parçalara bölünmesi gereksinimi ortadan kaldırılmıştır.

NSL-KDD veri setinde 4 farklı saldırı türü ve normal ağ trafiğine ait örnekler bulunmaktadır. Veri setinde bulunan saldırı türleri şu şekildedir: Hizmet dışı bırakma (DoS-Denial of service) , Bilgi tarama (Probe), Kullanıcının yönetici hesabına yükseltilmesi (User to Root-U2R) ve Uzaktan yerel oturum erişimi (Remote to Local Attack - R2L). NSL-KDD veri setinde farklı saldırı türlerinden oluşan DoS saldırısına ait 45927, U2R saldırısına ait 52, R2L saldırısına ait 995 ve PROBE saldırısına ait 11656 kayıt bulunmaktadır. 67343 adet kayıt ise normal veri trafiğine aittir. Veri seti, normal ve saldırı tiplerine ait toplamda 125973 kayıttan oluşmaktadır. Her bir kayıt, 41 farklı nitelik ve saldırı tipini gösteren 1 sınıf bilgisi birlikte 42 nitelik ile gösterilmektedir. Bu niteliklerde ağ trafiğine ait protokol tipi, servis bilgisi, bayraklar, paket bilgileri, sunucu ve kullanıcı ile ilgili birçok farklı bilgi bulunmaktadır. Tablo 1'de NSL-KDD veri setine ait özellik tablosu görülmektedir.

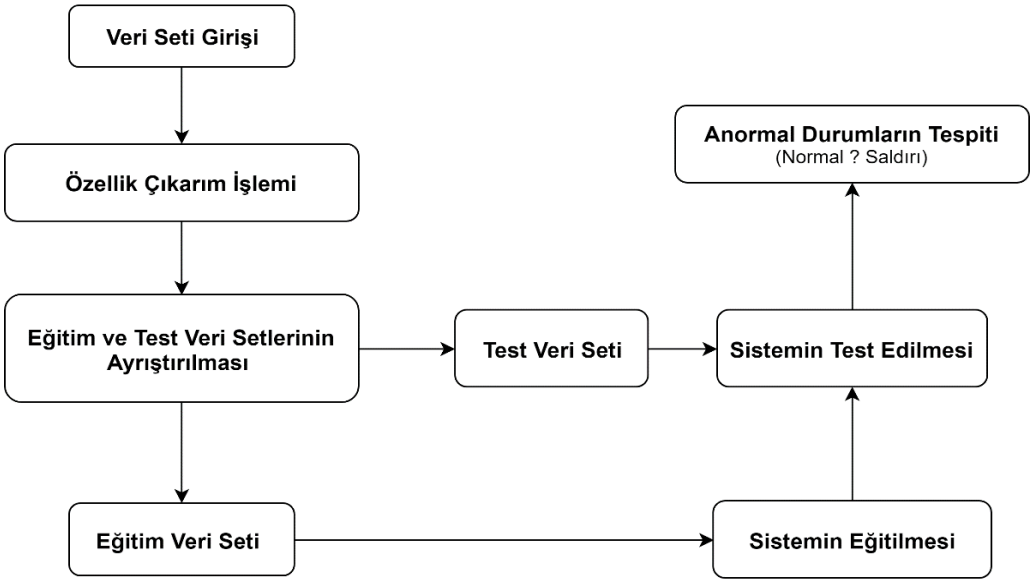
2.3. Weka

Weka [23] Waikato Üniversitesi tarafından geliştirilmiş açık kaynak bir veri madenciliği aracıdır. Weka kullanıcıları sunduğu grafik ara yüz sayesinde tüm işlemlerin çok daha kolay bir şekilde yapılmasını sağlamaktadır. Weka üzerinde veri setleri üzerinde işlem gerçekleştirmek için birçok paket barındırmaktadır. Veri setleri üzerinde önişlem yapılması, sınıflandırma ve kümeleme işlemleri için, birliktelik kurallarının tanımlanması, özellik çıkarımı ve görselleştirme işlemleri için ayrı paketler bulunmaktadır. Veri madenciliği uygulamalarında kullanılan daha birçok uygulama programı bulunmaktadır. Rapid Miner [24] ve R [25] bunlardan en yaygın olarak kullanılan programlar arasındadır. Bu programların birbirine göre avantaj ve dezavantajları bulunmaktadır.

3. ANAORMAL TRAFİK TESPİTİ TESTLERİ VE PERFORMANS DEĞERLENDİRMESİ

3.1. Anormal Trafik Tespiti

Bu bölümde ağ trafiğindeki anormal durumların tespit edilmesi için, farklı özellik çıkarım ve veri madenciliği algoritmalarının birlikte kullanıldığı sistemlerin testleri gerçekleştirilmiştir. Testlerin nasıl gerçekleştirildiği detaylı bir şekilde açıklanmıştır. Şekil 3'te testlerin nasıl gerçekleştirildiğini açıklamak için bir blok diyagram sunulmuştur. Şekil 3'te görüldüğü gibi, ilk olarak, NSL-KDD veri seti üzerinde özellik çıkarım işlemleri uygulanmaktadır. Bu aşamada farklı özellik çıkarım algoritmaları ile testler gerçekleştirilmiştir. Uygun bir özellik çıkarım algoritmasının kullanılması, veri seti üzerindeki başarımın artmasını ve işlem sürelerinin azalmasını sağlamaktadır. Bu çalışmada elde edilen ve Tablo 3'te verilen karşılaştırma sonuçlarında, performans sonuçlarının iyi ve daha kısa zamanda tamamlandığı gösterilmiştir.



Şekil 2. Veri madenciliği tabanlı anormallik tespit sistemi blok diyagramı

Tablo 1. NSL-KDD veri seti öznitelikleri

1	duration	**12**	logged_in	**23**	count	**34**	dst_host_same_srv_rate
2	protocol_type	**13**	num_compromised	**24**	srv_count	**35**	dst_host_diff_srv_rate
3	service	**14**	root_shell	**25**	serror_rate	**36**	dst_host_same_src_port_rate
4	flag	**15**	su_attempted	**26**	srv_serror_rate	**37**	dst_host_srv_diff_host_rate
5	src_bytes	**16**	num_root	**27**	rerror_rate	**38**	dst_host_serror_rate
6	dst_bytes	**17**	num_file_creations	**28**	srv_rerror_rate	**39**	dst_host_srv_serror_rate
7	land	**18**	num_shells	**29**	same_srv_rate	**40**	dst_host_rerror_rate
8	wrong_fragment	**19**	num_access_files	**30**	diff_srv_rate	**41**	dst_host_srv_rerror_rate
9	urgent	**20**	num_outbound_cmds	**31**	srv_diff_host_rate		
10	hot	**21**	is_host_login	**32**	dst_host_count		
11	num_failed_logins	**22**	is_guest_login	**33**	dst_host_srv_count		

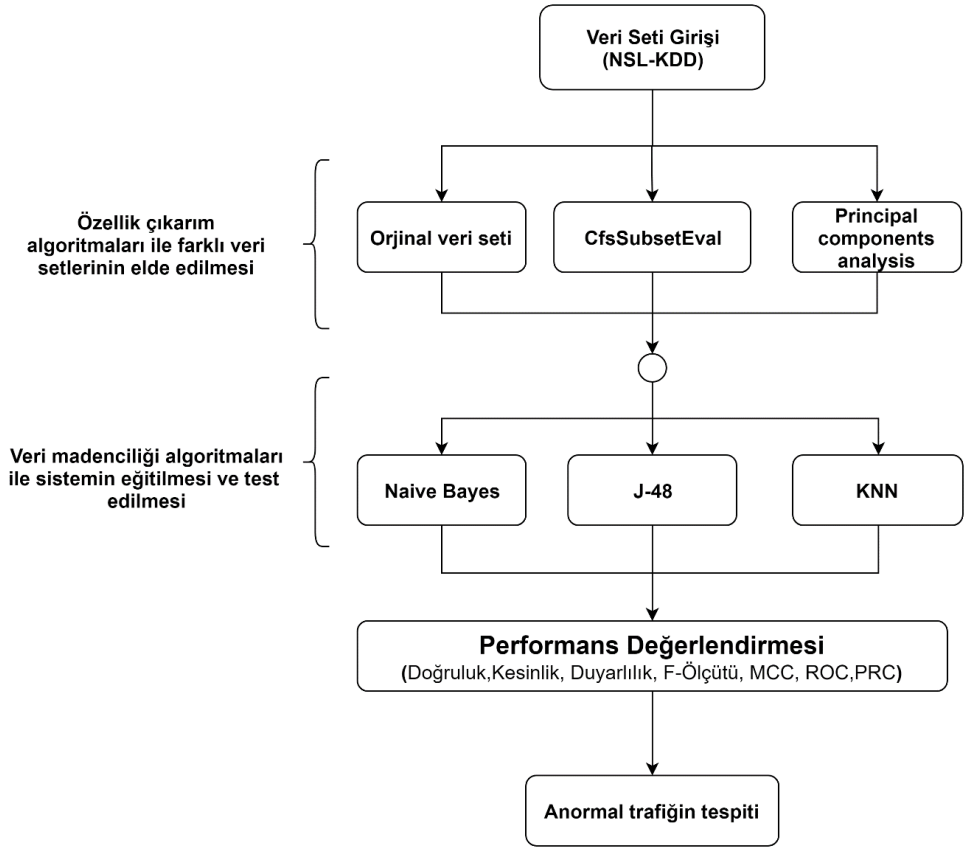
Bu işlemin ardından, NSL-KDD veri seti eğitim ve test setleri olarak 2'ye ayrılmaktadır. Bu çalışmada veri setinin %66,6'sı eğitim için %33,33'ü test için ayrılmıştır. Eğitim için ayrılan kısım üzerinden sistemin eğitimi gerçekleştirilmiş ve kalan kısım için test verisi olarak kullanılmıştır. Eğitim ve test verisinin ayrılmasında farklı yaklaşımlar bulunmaktadır. Fakat bu çalışmada test için kullanılan veri daha önceki hiçbir şekilde eğitim verisi olarak kullanılmamış ve sadece sistemin testinde kullanılmıştır. Bu şekilde anormal yani saldırı tiplerinden herhangi birine ait olan kayıtların tespiti hedeflenmektedir. Bu makalede Naive Bayes, J-48 Karar ağacı ve K en yakın komşu veri madenciliği algoritmaları ile test işlemleri yapılmış ve algoritmaların başarımları test edilmiştir.

3.2. Özellik Çıkarım Algoritmaları

Özellik çıkarım işlemi veri madenciliği uygulamalarında en önemli adımlardan birisidir. Uygun özellik seçim algoritmasının belirlenmesi ve işlemlerde kullanılması, uygulamanın sonuçlarında başarımı artıracak bir etkiye sahiptir. Ayrıca veri seti üzerindeki nitelik sayısını azalttığından veya nitelik arasındaki yeni bağlantıları tespit ettiğinden dolayı işlem yükünü hafifletecek bir etkiye de sahiptir[26-28]. Fakat özellik çıkarım işlemleri için kullanılan tek bir yöntem yoktur. Kullanılacak olan yöntem veri setinin durumuna göre çok değişkenlik gösterebilir. Özetle farklı veri setleri için farklı özellik çıkarım algoritmaları daha uygun olabilir. Özellik çıkarım yaklaşımında kullanılan birçok farklı algoritma bulunmaktadır [29-32]. Bu makalede farklı özellik çıkarım metotları denenerek, seçilen en uygun iki tanesi ile veri seti üzerinde işlemler gerçekleştirilmiştir. Bu algoritmalara ait kısa açıklamalar aşağıda bulunmaktadır.

CfsSubsetEval algoritması [33,34], veri setindeki niteliklerin yüksek derecede sınıf ile olan ilişkisini ve nitelikler arasındaki daha az önemli olanları belirleyerek, nitelikler arasında bir eleme gerçekleştirmektedir. Veri seti için en önemli olan özellikleri tespit edilmesini sağlamaktadır.

CfsSubsetEval bir arama algoritması kullanmaktadır. Algoritmada nitelikler arasından, sınıf etiketi ile en iyi ilişkiye sahip olanların belirlenmesi sağlanmaktadır. Bu durumda belirlenen özellik grubu sınıf etiketi ile yüksek bir bağlantı içermekte fakat diğer niteliklerin daha önemsiz bir duruma sahip olduğu tespit edilmektedir. Dolayısıyla sınıf etiketi ile yüksek bağlantıya sahip olan nitelik grubunun veri madenciliği işlemlerinde kullanımının başarımı artırması beklenmektedir. Bu çalışmada CfsSubSetEval algoritmasında BestFirst arama algoritması kullanılmıştır.



Şekil 3. Performans testleri blok diyagramı

Bu makalede veri seti üzerinde özellik çıkarımı için kullanılan bir diğer yöntem PCA (Principal components analysis) Temel bileşenler analizi yöntemidir [35,36]. PCA'da birçok niteliğe sahip veri setinin, daha az sayıda nitelik ve niteliklerinin birleşiminden elde edilen değerler ile ifade edilmesi sağlanmaktadır. PCA işlemi sonucunda elde edilen veri setinde daha az boyuta sahip bir yapı elde edilmektedir. Analizde veri setinde yer alan niteliklerin bir biri ile olan bağlantıları tespit edilmektedir. PCA' da işlemler veri setine ait kovaryans matrisinin çıkarılması ile gerçekleştirilmektedir. Elde edilen birden fazla boyutlu yapının bir biri ile olan ilişkisinin tespiti için kullanılmaktadır. Oluşturulan veri setindeki yeni nitelikler, farklı niteliklerin aralarındaki ilişkiye göre birleşiminden oluşmaktadır. PCA veri setindeki nitelikler arasında yüksek korelasyon bulunduğunda gereksiz verilerin elenmesi işlemini yerine getirmektedir. Bu analizde veri setindeki bazı özelliklerin kaybolması durumu söz konusudur fakat işlemler sırasında varyans değeri korunarak kayıpların en aza indirilmesi gerekmektedir.

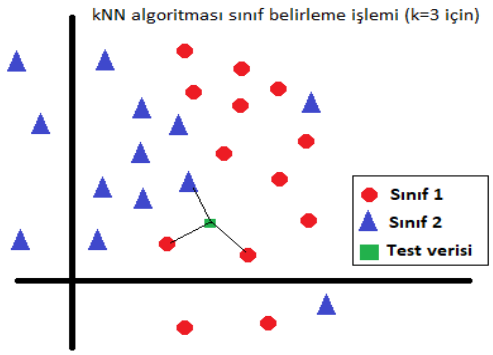
3.3. Veri Madenciliği Algoritmaları

Veri setleri üzerinde veri madenciliği işlemlerini gerçekleştirmek için birçok farklı veri madenciliği algoritması geliştirilmiştir. Bunlardan bazıları: Bayes [37], Naive Bayes[38], Bulanık mantık[39], Karar ağaçları[40], Yapay Sinir Ağları (YSA)[41] ve Karar Destek Makineleri (SVM) [42], K en yakın komşu (kNN)[43], K-ortalama [44]. Veri madenciliği algoritmaları veri setleri üzerinde çok farklı yaklaşımlar kullanarak işlemleri gerçekleştirmektedir. Bu algoritmaların başarımları üzerinde kullanıldıkları veri setlerinin yapısına göre farklılık göstermektedir. Yüksek oranda başarım için ilgili veri setinin yapısına uygun algoritma seçimi gerçekleştirilmelidir[45,46]. Bu bölümde NSL-KDD veri seti üzerinde sınıflandırma işlemleri için kullanılan veri madenciliği algoritmaları kısaca açıklanmıştır.

Naive Bayes sınıflandırma algoritması[38] Bayes teoreminin basitleştirilmiş halini kullanan bir algoritmadır. Algoritma hesaplamada koşullu olasılık kullanmaktadır. Eğitimi veri seti üzerinde sistem eğitilmekte ve test veri

setinin hangi sınıfa dahil olduğu tespit edilmeye çalışılmaktadır. Naive Bayes algoritması basit bir yapıya sahip olmasına rağmen, oldukça başarılı sonuçlar üretmektedir.

J-48 (C4.5) Karar Ağacı Algoritması literatürde C 4.5 [40] olarak bilinmektedir. ID3 olarak bilinen algoritmanın geliştirilmesi ile ortaya çıkmıştır. ID3 algoritmasından farklı olarak normalizasyon işlemlerinin bu algoritmaya dahil edilmesidir. Algoritmada bilgi kazancı değerleri hesaplanır ve bir oran olarak bu değerler kullanılmaktadır. Karar ağacının oluşturulmasında alt ağaçlar oluşturulur ve bu alt seviyedeki ağaçların farklı seviyelere taşınması mümkündür. Karar ağacı algoritmasında ayrıca veri setinde bulunan problemli verilerin ağaçtan silinmesi için dal budama işlemi gerçekleştirilmektedir. Ağaç oluşturma işleminde tek bir düğüm tespit edilerek işleme başlanır ve örneklerin tamamı tek bir sınıfta ise ilgili düğüm yaprak olarak tespit edilir ve bir sınıfı temsil eder. Eğer düğümde farklı sınıflara ait özellikler bulunuyorsa, en iyi bölümlemeyi yapacak özellik belirlenir ve dallanma devam eder. Bütün özelliklerin bilgi kazancı değerleri hesaplanır ve en iyi bilgi kazancı değerine sahip nitelik ağaçta karar düğümü olarak belirlenir. Belirlenen karar düğümünün altında yeni bir alt ağaç oluşturulmak suretiyle işleme devam edilir. Burada belirlenen alt gruplarda, bütün elemanlar için aynı değer elde ediliyorsa, ilerleme durur ve tespit edilen son değer çıkış değeri olarak belirlenir. Eğer alt grupta tek bir düğüm mevcutsa ve ayırt edici bir nitelik tespit edilemiyorsa ilerleme kesilir.



Şekil 4. kNN algoritması sınıf belirleme işlemi

kNN sınıflandırma algoritması [47] eğitimli öğrenme algoritmalarından en yaygın olarak kullanılanlarından birisidir. Büyük veri kümelerinde işlem süreleri diğer algoritmalara göre daha uzun sürebilir fakat başarılı sonuçlar üretmektedir. Algoritmada test edilecek olan yani sınıfı belirlenecek olan verinin eğitim kümesindeki her bir örneğe olan uzaklığı hesaplanmaktadır. Burada k sayısı sınıfın belirlenmesinde dikkate alınacak olan en yakın komşu sayısını ifade etmektedir. K değeri 1, 3, 5 gibi tek bir sayı olarak belirlenir ve hesaplanan en yakın komşuların dahil olduğu sınıfa göre yeni örneğin sınıfı belirlenmektedir. K değerinin yüksek veya düşük seçilmesi bazı durumlarda sonuçları olumsuz etkilemektedir. Üzerinde çalışılan veri setine göre uygun k değerinin belirlenmesi oldukça önemlidir. Şekil 4'te kNN algoritmasında yeni bir örneğe ait sınıf belirleme işlemi görülmektedir. Örnekte K=3 değeri için hesaplama yapılmış ve bu düğüme en yakın 3 komşu belirlenmiştir. Belirlenen en yakın 3 komşudan 2 tanesi sınıf 1'e ait olduğu için test düğümününse sınıf 1'e ait olduğu bu işlem sonucunda belirlenmiştir.

3.4. Performans Testleri

Bu bölümde, farklı özellik çıkarım ve veri madenciliği algoritmaları ile testler gerçekleştirilmiştir. NSL-KDD veri seti üzerinde yüzde ayrım kullanılarak eğitim ve test veri setleri ayrıştırılmıştır. Veri seti üzerinde ilk olarak farklı özellik çıkarım algoritmaları kullanılarak veri seti özelleştirilmiştir. Cfssubeval algoritması kullanılarak, 42 nitelik 7 niteliğe indirgenerek veri seti oluşturulmuştur. PCA algoritması ile de 42 nitelikten 6, 11 ve 21 niteliğe sahip 3 farklı veri üretilmiştir. Bu değerler orijinal veri setinin nitelik sayısının dörtte biri ve yarı sayısına eşit olacak şekilde belirlenmiştir. İki özellik çıkarımı ile elde edilen veri setlerinin yanında, hiçbir özellik çıkarımı işlemi uygulanmaksızın orijinal veri seti ile de testler yapılmıştır. Özellik çıkarımı ile oluşturulan ve orijinal veri setleri üzerinde, Naive Bayes, J48 ve KNN (1-3) veri madenciliği algoritmaları ile testler gerçekleştirilmiştir. Algoritmaların işlem zamanları belirlenmiştir. Tüm testlerde NSL-KDD veri setinin %33,33 olan 42521 örnek üzerinde test yapılmış ve bu değer üzerinden değerlendirme yapılmıştır. Veri setinin kalan kısmı (% 66,66) eğitim için kullanılmıştır. Testler sonucunda önerilen sistem için hangi algoritmaların daha iyi sonuçlar ürettiği ortaya konulmuştur.

Önerilen sistemin NSL-KDD veri üzerindeki başarımını tespit etmek için aşağıda açıklanan değerlendirme kriterleri [48,49,50] kullanılmıştır. Değerlendirme kriterlerinde kullanılmak üzere ilk başta Tablo 2'de görülen karmaşıklık matrisi elde edilmiştir. Karmaşıklık matrisi üzerinden değerlendirme kriterleri elde edilmiştir.

Tablo 2. Karmaşıklık matrisi

		Tahmin edilen	
		Saldırı	Normal
Gerçek	Saldırı	DP (doğru-pozitif)	YN (yanlış-negatif)
	Normal	YP (yanlış-pozitif)	DN (doğru-negatif)

Karmaşıklık matrisinde hesaplanan değerlerin açıklaması şu şekildedir:

DP (doğru-pozitif): Veri setinde saldırı sınıfında olup, saldırı sınıfında tahmin edilen örnek sayısı.
YN (yanlış-negatif): Veri setinde saldırı sınıfında olup, normal olarak tahmin edilen örnek sayısı.
YP (yanlış-pozitif): Veri setinde normal sınıfında olup, saldırı olarak tahmin edilen örnek sayısı.
DN (doğru-negatif): Veri setinde normal sınıfta olup, normal olarak tahmin edilen örnek sayısı.

Karışıklık matrisi elde edildikten sonra, bu değerler kullanılarak hesaplanan değerlendirme kriterlerin açıklaması aşağıda verilmiştir.

Doğruluk: Bir algoritmasının doğru olarak sınıflandırma yaptığı örnek sayısının, toplam örnek sayısına oranı bu değeri vermektedir. Bu değerin hesaplanmasında veri setinden test için ayrılan kısmın sayıları kullanılmaktadır.

Doğruluk= DP+DN/ DP+YN+YP+DN (1)

Duyarlılık: Veri setinde saldırı sınıfında olan ve saldırı olarak tahmin edilen örnek sayısının, gerçekte saldırı olan tüm örneklere oranı duyarlılık değerini vermektedir.

Duyarlılık= DP/DP+YN (2)

Kesinlik: Veri setinde saldırı sınıfında olup saldırı olarak sınıflandırılan değerlerin sayısının, saldırı olarak tahmin edilen tüm örneklerin sayısına oranıdır.

Kesinlik= DP/ DP+YP (3)

F-Ölçütü: Bu değerlendirme kriterinde kesinlik ve duyarlılık değerleri birlikte kullanılarak yeni bir değer hesaplanmaktadır. Bu değer elde edilen kesinlik ve duyarlılık değerlerin harmonik ortalaması hesaplanarak elde edilmektedir.

F-Ölçütü=2x(kesinlik x duyarlılık/ kesinlik + duyarlılık) (4)

MCC (Matthews Correlation Coefficients**):**

MCC[51] özelikle iki sınıfı bulunan veri setleri üzerinde yapılan işlemlerde diğer performans kriterlerine göre daha doğru sonuçlar üretmektedir. Özellikle dengesiz olarak dağılmış veri setleri üzerinde bile en gerçekçi sonucun elde edilmesini sağlamaktadır. MCC değeri aşağıda belirtildiği gibi hesaplanmaktadır. Hesaplanan değerin 1'e yakın olması doğru bir sınıflandırma yapıldığını göstermektedir.

$$MCC = \frac{(DPxDN) - (YPxYN)}{\sqrt{(DP+YP)x(DP+YN)x(DN+YP)x(DN+YN)}} \quad (5)$$

ROC:

ROC analizi [52] veri setinde üzerinde gerçekleştirilen sınıflama işleminin performans ölçümünde kullanılan bir diğer önemli analizdir. Bu analizde sınıflandırma işleminin hangi oranda doğru tahminde bulunduğu farklı bir yaklaşım kullanılarak incelenmektedir. ROC eğrisinin elde edilmesinde Duyarlılık ve Özgüllük ile ifade edilen iki kavram kullanılmaktadır. Y ekseninde hesaplanan duyarlılık değeri X ekseninde ise hesaplanan (1- Özgüllük) değeri yer almaktadır. Bu iki değerin kesişimleri ile elde edilen noktaların birleşimi ile ROC eğrisi elde edilmektedir.

Özgüllük=DN/(DN+YP) (6)

ROC analizinden elde edilen değer, ROC eğrisinin altında kalan alanın büyüklüğünü göstermektedir. ROC testinde 1 en iyi değeri, 0,5 ise başarısız bir sınıflandırma yapıldığını göstermektedir. ROC değeri 1 olduğunda veri seti üzerinde hiçbir yanlış tahmin yapılmadığı anlaşılmaktadır ve bu ROC eğrisi koordinat düzleminde (0,0), (1,0) ve (1,1) noktalarını birleştiren bir doğru çizmektedir. ROC eğrisi y=x fonksiyonuna eğilim gösterdiği kadarıyla başarısız bir işlem gerçekleştirildiğini ortaya koymaktadır.

PRC (Precision-recall Curve):

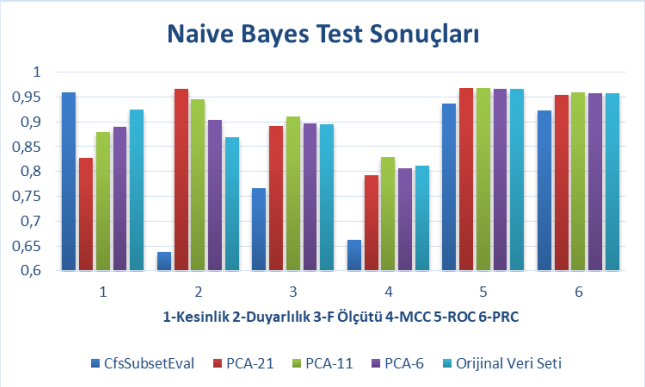
Bu performans kriterinde iki önemli performans değeri olan duyarlılık ve kesinlik arasında ilişki irdelenmektedir. Her bir kesim noktasında bu iki değer hesaplanarak bir eğri elde edilmektedir. Eğrinin çizilmesinde X ekseninde hesaplanan duyarlılık değeri ve Y ekseninde ise kesinlik değeri bulunmaktadır. Bu iki değerin kesişiminden elde edilen noktaların birleşimi ile PRC eğrisi elde edilmektedir. Bu performans kriterinde de 1'e yakın değerlerin elde edilmesi ne kadar doğru bir tahmin yapıldığına ilişkin bir bilgi sağlamaktadır.

x ekseninde → Duyarlılık
y ekseninde → Kesinlik

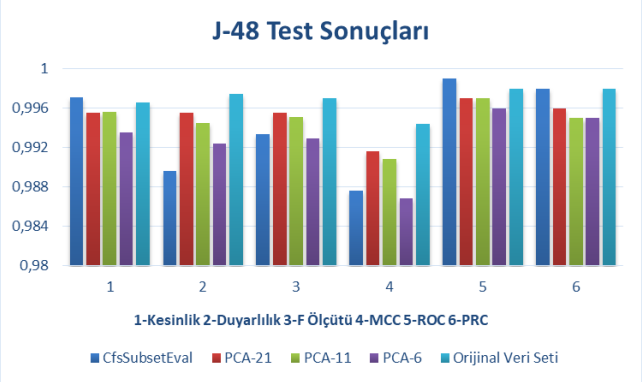
Tablo 3'te farklı özellik çıkarım algoritması ve orijinal veri setleri üzerinde, farklı veri madenciliği algoritmaları kullanılarak gerçekleştirilen testlere ait sonuçlar görülmektedir. Performans kriterleri olarak, doğruluk, kesinlik, duyarlılık, F-ölçütü, MCC, ROC, PRC ve işlem zamanlarına ait değerler tabloda sunulmuştur.

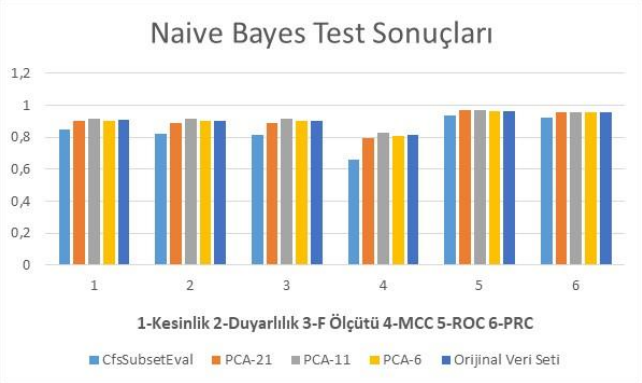
3.4.1. Performans Testleri Sonuçları

Weka'da farklı veri madenciliği ve özellik çıkarım algoritmaları kullanılarak yapılan test sonuçları Tablo 3'te verilmiştir. Test sonuçları incelendiğinde, en yüksek doğruluk oranının orijinal veri seti üzerinde J48 karar ağacı kullanılarak elde edildiği görülmektedir. Tüm veri setleri ve tüm algoritmalar üzerinde Naive Bayes algoritmasının en düşük doğruluk oranına sahip olduğu fakat en düşük işlem zamanına sahip olduğu tespit edilmiştir. KNN algoritmasında 1 ve 3 en yakın komşulukları için gerçekleştirilen testlerde, 1 komşu için doğruluk oranının daha iyi sonuçlar verdiği görülmektedir. CfsSubsetEval özellik çıkarım algoritması ile nitelik sayısının 42'den 7'ye düşürülmesi ile indirgenmiş veri seti üzerinde yapılan işlemlerde, Naive Bayes ve J48 karar ağacı algoritmalarının işlem sürelerinin oldukça düşük ve hızlı işlem gerçekleştirdikleri ve Naive Bayes algoritmasında performans değerlerinin düşük olmasına karşın, J48 karar ağacı algoritmasında iyi performans değerlerine sahip olduğu belirlenmiştir.

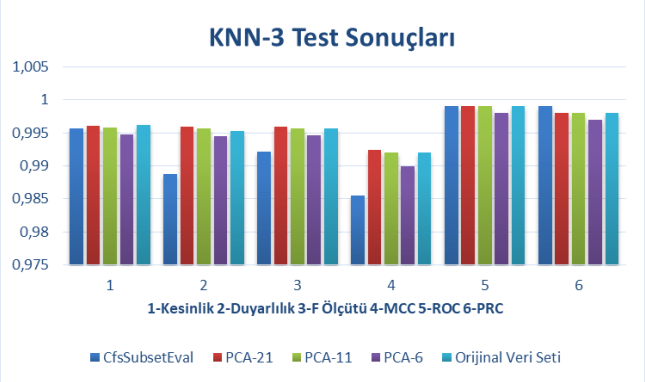


Şekil 5. Naive Bayes Algoritması Test Sonuçları

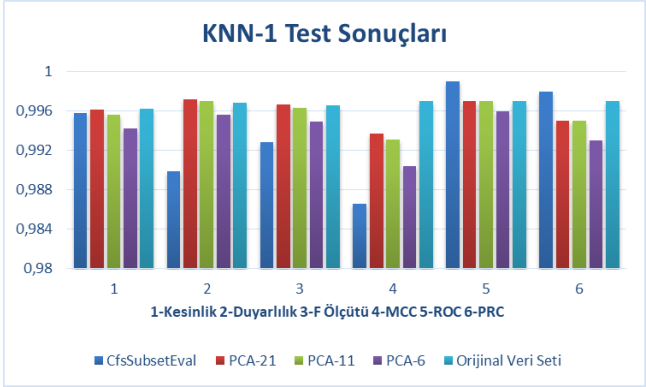




Şekil 6. J-48 Algoritması Test Sonuçları



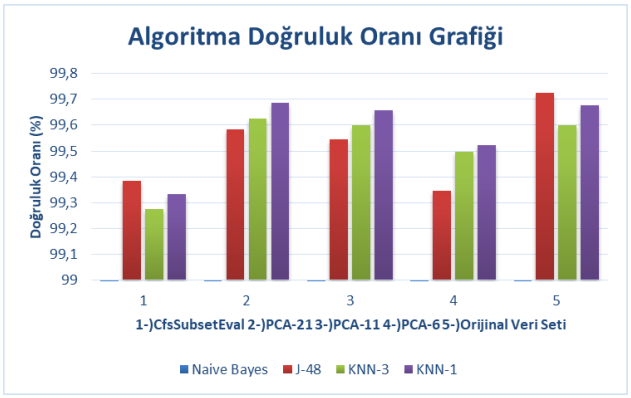
Şekil 7. KNN Algoritması Test Sonuçları (n=3)



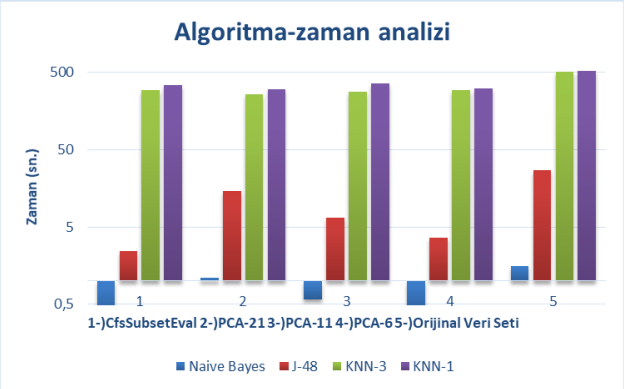
Şekil 8. KNN Algoritması Test Sonuçları(n=1)

PCA özellik çıkarım algoritmasında ise nitelik sayısı farklı değerler üzerinden değerlendirme yapılmıştır. PCA analizi sonucu belirlenen 6, 11 ve 21 nitelik üzerinden testler yapılmıştır. PCA analizi sonucu elde edilen test değerlerinin CfsSubsetEval algoritması kullanılarak elde edilen veri setinden genel olarak daha iyi sonuçlar elde ettiği görülmektedir.

Kesinlik, Duyarlılık, F-Ölçütü, MCC, ROC ve PRC kriterlerine göre Tablo 3'teki sonuçlar incelendiğinde PCA (21 nitelik) ve orijinal veri seti üzerinde KNN(1) algoritması kullanılarak elde edilen sonuçların diğerlerinden iyi olduğu tespit edilmiştir. Performans sonuçları, işlem zamanı ile birlikte değerlendirildiğinde ise CfsSubsetEval özellik çıkarım algoritması ile edilen veri seti üzerinde J-48 algoritması ile yapılan işlemlerin hem yüksek başarıma hem de çok iyi bir işlem zamanına sahip olduğu görülmektedir. KNN algoritmasının ise diğer algoritmalara kıyasla oldukça yüksek işlem zamanı gerektirdiği tespit edilmiştir. Şekil 5, 6, 7 ve 8'de farklı veri madenciliği algoritmaları ile gerçekleştirilen testlere ait Kesinlik, Duyarlılık, F-Ölçütü, MCC, ROC ve PRC test sonuçları görülmektedir. Özellik çıkarım algoritmaları ile elde edilen farklı veri setlerine ait sonuçlarda şekillerde görülmektedir.



Şekil 9. Algoritma-Doğruluk Oranı Grafiği



Şekil 10. Algoritma- Zaman Analiz Grafiği

Tablo 3. Performans karşılaştırma tablosu

Algoritmalar		Karmaşıklık matrisi		Doğruluk (%)	Kesinlik	Duyarlılık	F-Ölçütü	MCC	ROC	PRC	Time (sn.)
CfsSubsetEval (7 nitelik)	Naive Bayes	1273	7233	81,8543	0,9594	0,6378	0,7662	0,6625	0,937	0,923	0,31
		539	2232								
	J-48	1976	206	99,3860	0,9971	0,9897	0,9934	0,9877	0,999	0,998	2,47
		57	2280								
	KNN-3	1974	224	99,2762	0,9957	0,9888	0,9922	0,9855	0,999	0,999	296,85
		86	2277								
	KNN-1	1976	203	99,3323	0,9958	0,9898	0,9928	0,9866	0,999	0,998	338,22
		83	2277								
PCA (21 nitelik)	Naive Bayes	1931	652	89,0920	0,8277	0,9673	0,8921	0,7929	0,968	0,955	1,1
		4020	1884								
	J-48	1988	89	99,5844	0,9955	0,9955	0,9955	0,9917	0,997	0,996	14,48
		89	2277								
	KNN-3	1988	82	99,6241	0,9960	0,9959	0,9960	0,9924	0,999	0,998	261,39
		79	2278								
	KNN-1	1991	57	99,6871	0,9961	0,9971	0,9966	0,9937	0,997	0,995	302,65
		77	2278								
PCA (11 nitelik)	Naive Bayes	1887	1097	91,3847	0,8792	0,9451	0,9109	0,8297	0,969	0,959	0,58
		2593	2026								
	J-48	1986	109	99,5447	0,9957	0,9945	0,9951	0,9909	0,997	0,995	6,67
		86	2277								
	KNN-3	1988	87	99,6008	0,9958	0,9956	0,9957	0,9920	0,999	0,998	278,95
		84	2277								
	KNN-1	1990	60	99,6568	0,9956	0,9970	0,9963	0,9931	0,997	0,995	357,49
		87	2277								
PCA (6 nitelik)	Naive Bayes	1806	1907	90,3341	0,8900	0,9045	0,8972	0,8061	0,967	0,958	0,36
		2233	2062								
	J-48	1981	152	99,3463	0,9936	0,9924	0,9930	0,9869	0,996	0,995	3,6
		128	2273								
	KNN-3	1985	110	99,4980	0,9947	0,9945	0,9946	0,9899	0,998	0,997	291,49
		105	2275								
		1988	88								

	KNN-1	116	2274	99,5237	0,9942	0,9956	0,9949	0,9904	0,996	0,993	305,92
Orijinal veri seti (42 nitelik)	Naive Bayes	1735	2615	90,5933	0,9247	0,8690	0,8960	0,8116	0,966	0,958	1,57
		1414	2144								
	J-48	1991	50	99,7245	0,9966	0,9975	0,9970	0,9945	0,998	0,998	27,28
		68	2279								
	KNN-3	1987	95	99,6008	0,9962	0,9952	0,9957	0,9920	0,999	0,998	508,72
		76	2278								
	KNN-1	1990	63	99,6778	0,9962	0,9968	0,9965	0,9970	0,997	0,997	518,76
		75	2278								

4. SONUÇ VE DEĞERLENDİRME

Bu makalede, bilgisayar ağları üzerinde zararlı içeriğe sahip trafiğin tespit edilmesi ve saldırıların algılanabilmesi için, saldırı tespit sistemlerinde kullanılan veri madenciliği algoritmalarının performans değerlendirmesi yapılmıştır. Testleri gerçekleştirmek için NSL-KDD veri seti seçilmiş olup, bu veri seti üzerinde özellik seçim algoritmaları ile veri belirlenen farklı niteliklerden oluşan veri setleri elde edilmiştir. Farklı özellik çıkarım algoritmaları ile oluşturulan veri setleri üzerinde farklı veri madenciliği algoritmaları ile testler yapılmıştır. Elde edilen performans test sonuçları incelenmiş ve en iyi sonuçların elde edildiği algoritmalar belirlenmiştir. Şekil 9'da farklı veri setleri üzerinde, farklı algoritmalar ile gerçekleştirilen testler sonucunda elde edilen doğruluk oranları sunulmuştur. Şekil 9 incelendiğinde, en yüksek doğruluk oranının PCA 21 nitelikli ve orijinal veri seti üzerinde kNN-1 ve J-48 algoritmaları ile elde edildiği görülmektedir.

Şekil 10'da ise elde edilen işlem zamanları logaritmik olarak gösterilmiştir. Şekil 10'da görüldüğü gibi Naive Bayes ve J-48 algoritmasının işlem zamanı kNN algoritmasına göre oldukça düşüktür. kNN algoritması hem n=1 için hem de n=3 için oldukça yüksek işlem zamanlarına sahiptir. Test sonuçlarına göre, J-48 algoritmasının CfsSubsetEval özellik çıkarım algoritması ile elde edilen veri setinde, hem işlem zamanı hem de başarım oranları açısından oldukça iyi olduğu tespit edilmiştir.

KAYNAKÇA

[1] R. Deng, P. Zhuang, and H. Liang, "CCPA: Coordinated Cyber-Physical Attacks and Countermeasures in Smart Grid," IEEE Transactions on Smart Grid, vol. 8, no. 5, pp. 2420–2430, 2017.

[2] S. Wang, A. Zhou, M. Yang, L. Sun, C.-H. Hsu, and F. Yang, "Service Composition in Cyber-Physical-Social Systems," IEEE Transactions on Emerging Topics in Computing, pp. 1–1, 2017.

[3] L. Qi, W. Dou, Y. Zhou, J. Yu, and C. Hu, "A Context-aware Service Evaluation Approach over Big Data for Cloud Applications," IEEE Transactions on Cloud Computing, pp. 1–1, 2015.

[4] D. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, 1987.

[5] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating Computer Intrusion Detection Systems," ACM Computing Surveys, vol. 48, no. 1, pp. 1–41, 2015.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, Jan. 2009.

[7] K. Julisch, "Data Mining for Intrusion Detection," Advances in Information Security Applications of Data Mining in Computer Security, pp. 33–62, 2002.

[8] W. Lee, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344).

[9] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs decision trees in intrusion detection systems," Proceedings of the 2004 ACM symposium on Applied computing - SAC 04, 2004.

[10] M. Blowers and J. Williams, "Machine Learning Applied to Cyber Operations," Advances in Information Security Network Science and Cybersecurity, pp. 155–175, 2013.

[11] G. Nadiammai and M. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques," Egyptian Informatics Journal, vol. 15, no. 1, pp. 37–50, 2014.

[12] J. Patel, K. Panchal, "Effective Intrusion Detection System using Data Mining Technique", Journal of Emerging Technologies and Innovative Research (JETIR), Vol. 2, no. 6, pp 1869- 1878, 2015.

[13] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," Expert Systems with Applications, vol. 39, no. 1, pp. 424–430, 2012.

[14] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and

support vector machines," Expert Systems with Applications, vol. 38, no. 1, pp. 306–313, 2011.
[15] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-Based Systems, vol. 78, pp. 13–21, 2015.
[16] C.-B. Jiang, I.-H. Liu, Y.-N. Chung, and J.-S. Li, "Novel intrusion prediction mechanism based on honeypot log similarity," International Journal of Network Management, vol. 26, no. 3, pp. 156–175, Dec. 2016.
[17] B. K. Kumar, A. Bhaskar, "Identifying Network Anomalies Using Clustering Technique in Weblog Data", International Journal of Computers & Technology, Vol. 2 No. 3,2012.
[18] V. Sharma and A. Nema, "Innovative Genetic Approach for Intrusion Detection by Using Decision Tree," 2013 International Conference on Communication Systems and Network Technologies, 2013.
[19] A. Ashoor, S. Gore , "Intrusion Detection System (IDS): Case Study", International Conference on Advanced Materials Engineering,2011.
[20] J. Stenico, L. Ling, "Network Traffic Monitoring and Analysis", The State of the Art in Intrusion Prevention and Detection, 23-46, 2014.
[21] The KDD CUP 1999 Data. 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Erişim zamanı; Nisan, 2, 2018)
[22] NSL-KDD, URL: http:// unb.ca/cic/datasets/nsl.html (Erişim zamanı; Nisan, 3, 2018)
[23] M. Hall, E. Frank, J. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: An update," ACM SIGKDD Explor. Newslett., vol. 11, no. 1, pp. 10–18, 2009.
[24] R Language Definition, R Core Team, URL: ftp://155.232.191.133/cran/doc/manuals/r-devel/R-lang.pdf (Erişim zamanı; Nisan, 3, 2018)
[25] M. Graczyk, T. Lasota, and B. Trawiński, "Comparative Analysis of Premises Valuation Models Using KEEL, RapidMiner, and WEKA," Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems Lecture Notes in Computer Science, pp. 800–812, 2009.
[26] M. A. Hall, L. A. Smith, "Practical feature subset selection for machine learning". In C. McDonald(Ed.), Computer Science '98 Proceedings, pp. 181-191,1998.
[27] H,Almuallim, T. G. Dietterich, "Efficient algorithms for identifying relevant features". In Proc. of the 9th Canadian Conference on Artificial Intelligence, pp. 38-45, 1991.
[28] K. Kenji, L. A. Rendell. "The feature selection problem: Traditional methods and a new algorithm." AAAI'92 Proceedings of the tenth national conference on Artificial intelligence, pp. 129-134. 1992.
[29] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," Machine Learning Proceedings 1994, pp. 121–129, 1994.
[30] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, no. 1-2, pp. 273–324, 1997.
[31] D. Sanmay. "Filters, wrappers and a boosting-based hybrid for feature selection." In ICML, vol. 1, pp. 74-81. 2001.
[32] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - a filter solution," 2002 IEEE International Conference on Data Mining, 2002. Proceedings.
[33] T. S. Chou, K. K. Yen, and J. Luo,"Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms", International Journal of Computational Intelligence, vol. 4, no. 3, pp.196-208,2008.
[34] K. Selvakuberan, M. Indradevi, Dr. R. Rajaram "Combined Feature Selection and classification – A novel approach for the categorization of web pages", Journal of Information and Computing Science, Vol. 3, No. 2, pp. 83-89, 2008.
[35] H. Abdi and L. J. Williams, "Principal component analysis," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, pp. 433–459, 2010.
[36] T. Metsalu and J. Vilo, "ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap," Nucleic Acids Research, vol. 43, no. W1, Dec. 2015.
[37] S. L. Scott, "A Bayesian paradigm for designing intrusion detection systems," Computational Statistics & Data Analysis, vol. 45, no. 1, pp. 69–83, 2004.
[38] D. Mladenic, M. Grobelnik, "Feature selection for unbalanced class distribution and naive bayes", In ICML Vol. 99, pp. 258-267,1999.
[39] K. Alsubhi, I. Aib, and R. Boutaba, "FuzMet: a fuzzy-logic based alert prioritization engine for intrusion detection systems," International Journal of Network Management, vol. 22, no. 4, pp. 263–284, 2011.
[40] R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[41] J. Cannady, "Artificial neural networks for misuse detection", In: Proceedings of the National Information Systems Security Conference; 368-381,1998.
[42] Z. Zhang and H. Shen, "Application of online-training SVMs for real-time intrusion detection with different considerations," Computer Communications, vol. 28, no. 12, pp. 1428–1442, 2005.
[43] T. Denœux, "A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory," IEEE transactions on systems, man, and cybernetics, vol. 25, no.5, pp. 804-813, 1995.
[44] J. A. Hartigan, M. A. Wong, "A k-means clustering algorithm. Journal of the Royal Statistical Society", Series C (Applied Statistics), vol. 28, no. 1, pp.100-108, 1979.
[45] E. Alpaydin, "Introduction to machine learning,"MIT Press, 2004.
[46] J. Han, and M. Kamber, "Data mining: concepts and techniques'" (2nd ed.). Morgan Kaufmann Publishers, 2006.
[47] Y. Liao and V. Vemuri, "Use of K-Nearest Neighbor classifier for intrusion detection," Computers & Security, vol. 21, no. 5, pp. 439–448, 2002.
[48] N. Japkowicz, M. Shah, "Evaluating learning algorithms: a classification perspective", Cambridge University Press, 2011.

[49] T. R. Patil, S. S. Sherekar, "Performance analysis of Naive Bayes and J48 classification algorithm for data classification", International Journal of Computer Science and Applications, vol. 6, no. 2, pp.256-261, 2013.
[50] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," Information Sciences, vol. 340-341, pp. 250–261, 2016.
[51] Y. Liu, J. Cheng, C. Yan, X. Wu, and F. Chen, "Research on the Matthews Correlation Coefficients Metrics of Personalized Recommendation Algorithm Evaluation," International Journal of Hybrid Information Technology, vol. 8, no. 1, pp. 163–172, 2015.
[52] J. A. Swets, "ROC Analysis Applied to the Evaluation of Medical Imaging echniques," Investigative Radiology, vol. 14, no. 2, pp. 109–121, 1979.