Makine Öğrenmesi ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti

Oğuz KAYNAR[1], Halil ARSLAN[2], Yasin GÖRMEZ[1], Yunus Emre IŞIK[1]

[1]Yönetim Bilişim Sistemleri Bölümü, Cumhuriyet Üniversitesi, Sivas Türkiye
[2]Bilgisayar Mühendisliği, Cumhuriyet Üniversitesi, Sivas Türkiye
okaynar@cumhuriyet.edu.tr, harslan@cumhuriyet.edu.tr, yasingormez@cuhmuriyet.edu.tr, yeisik@cuhmuriyet.edu.tr
(Geliş/Received: 19.12.2017; Kabul/Accepted: 15.03.2018)
DOI: 10.17671/gazibtd.368583

Özet— Bilgisayar ve internetin, günlük yaşamın vazgeçilmez bir unsuru haline gelmesi ile birlikte internet sitelerinin ve web tabanlı uygulamaların sayısı da hızla artmıştır. Bilgi, fikir, para gibi birçok önemli unsurun internet siteleri ve uygulamalar aracılığıyla paylaşımının yapılması ise bilgi güvenliği konusunu önemli ve güncel bir hale getirmiştir. Günümüze kadar güvenlik duvarı, virüs programları gibi yazılımlar bilgisayar ve sistem güvenliği için kullanılmış ancak yeterli olmamıştır. Bu nedenle mevcut yazılımlara alternatif olarak ortaya atılan saldırı tespit sistemleri ile anormal davranışlar tespit edilerek olası tehlikelerin çözümlenmesi amaçlanmıştır. Bu çalışmada ise saldırı tespit sistemleri için üretilen KDDCup99 veri seti üzerinde ki kare, bilgi kazancı, kazanım oranı, gini katsayısı, oneR, reliefF, genetik, ileriye doğru ve geriye doğru öznitelik seçim algoritmaları uygulanarak yeni veri setleri elde edilmiştir. Elde edilen yeni veri setlerini, orijinal boyuttaki veri seti ile karşılaştırmak için en yakın k komşu, destek vektör makineleri ve aşırı öğrenme makineleri kullanılarak farklı modeller geliştirilmiştir. Çalışmada, her üç yöntem için belirtilen öznitelik seçme yöntemleri kullanılarak test verileri için en yüksek başarıma sahip modeller başarı oranı, hassasiyet yanlış alarm oranı, f-ölçütü gibi çeşitli metrikler yardımıyla karşılaştırılmıştır. Yapılan analizlerin sonucunda öznitelik seçim yöntemlerinin her üç sınıflama yöntemi içinde başarı oranını artırdığı ve modellerin daha hızlı çalışmasını sağladığı görülmüştür. Ayrıca, yüksek başarı oranları, diğer sınıflama yöntemlerine oranla son derece hızlı olması, eğitim algoritmasının basit olması gibi nedenlerden dolayı aşırı öğrenme makinalarının çevrimiçi saldırı tespit sistemlerine rahatlıkla entegre edilebileceğini ve alternatif bir yöntem olarak kullanılabileceğini göstermiştir.

Anahtar Kelimeler— Saldırı Tespiti, Makine Öğrenmesi, Öznitelik Seçimi, Aşırı Öğrenme Makineleri

Intrusion Detection with Machine Learning and Feature Selection Methods

Abstract— As computers and the internet become indispensable elements of everyday life, the number of internet sites and web based applications have increased rapidly. The sharing of information, ideas and money through internet sites and applications has made information security an important and actual issue. Softwares such as security walls and virus programs are used for computer and system security have not been enough. For this reason, it is aimed to solve the possible threats by detecting abnormal behaviors with the intrusion detection systems that are proposed as an alternative to existing software. In this study, new data sets are obtained by applying chi square, information gain, gain ratio, gini coefficient, oneR, reliefF, genetic, forward and backward feature selection algorithms on KDDCup99 dataset generated for intrusion detection systems. In order to compare the new data sets obtained with the original size data set, different models are developed by using k-nearest-neighbor, support vector machines and extreme learning machines. In this study, all three methods for test data having best results are compared by using mentioned feature selection methods according to metrics such as success rate, precision, false alarm error, f-measure. As the result of experiments, it is seen that feature selection methods increases the success rate and enables models to function faster for all three classification methods. Besides, it shows that extreme learning machines are able to be integrated to intrusion detection systems conveniently and be used as an alternative method because of reasons such as having high success rate, being faster than other classification methods and having simple training algorithm.

Keywords— Intrusion Detection, Machine Learning, Feature Selection, Extreme Learning Machine

1. GİRİŞ (INTRODUCTION)

Son yıllarda teknolojik değişimler sayesinde internet artık insan hayatı için bir gereksinim haline gelmiştir. Geliştirilen internet siteleri ve uygulamalar sayesinde alışveriş, bankacılık işlemleri, haberleşme, eğlence gibi insan hayatı için vazgeçilmez olan; kişisel bilgi, para, fikir gibi önemli unsurların kullanıldığı birçok iş, bilgisayarlar ve mobil cihazların yardımı ile internet üzerinden yapılmaktadır. Bu uygulama ve internet siteleri insanlar için kolaylık sağlamaktadır ancak bu uygulamalardaki bazı güvenlik açıklıkları yüzünden maddi manevi kayıplar yaşanabilmektedir. Bu nedenle uygulamaların kullanışlı olması kadar güvenli olması da son derece önemli bir hal almıştır. Uygulamaların güvenliklerinin test edildiği, kötü yazılım içeren uygulamaların tespit edilerek silindiği, kötü niyet içeren bağlantıların tespit edilerek reddedildiği güvenlik duvarı ve virüs programları gibi yazılımlar bu kayıpları büyük ölçüde engellemektedir ancak olası kayıpların büyüklüğü göz önüne alındığında bu yazılımlar yeterli olmamaktadır. Bu duruma çözüm olarak sunulan Saldırı Tespit Sistemleri (STS), bir ağın ya da sistemin yaptığı aktiviteleri kontrol ederek saldırıları tespit etmeye çalışan, engellemek için karşı girişimde bulunmayan gerçek zamanlı çalışan yazılım ürünleridir.

STS temelinde bir tetikleme mekanizmasına gereksinim duymaktadır. Bu mekanizma ağ ya da sistem kullanımında normal olmayan, yanlış veya izin dışı yapılan, saldırı olarak nitelendirilen aktiviteler ile karşılaşması durumunda alarm veren yazılımlardır. Bu yazılımlarda saldırı olan aktiviteleri tespit etmek için imza tabanlı ve anormallik tabanlı olmak üzere iki farklı analiz yöntemi kullanılmaktır. İmza tabanlı STS'ler gelen bağlantıdan elde edilen bilgileri önceden oluşturulmuş imza veri tabanı ile karşılaştırarak saldırıyı tespit etmeye çalışır [1]. Bu imza veri tabanları önceden karşılaşılan saldırıların çeşitli bilgileri (Host IP, port numarası, paket bilgileri vb.) kullanılarak oluşturulmuştur ve yüksek boyuta sahip olabilmektedir. İmza tabanlı STS'lerin avantajı çok fazla yanlış alarm (saldırı olmayan bir bağlantının saldırı olarak nitelendirilmesi) vermemesidir. Dezavantajları ise büyük veri tabanlarına ihtiyaç duyması ve veri tabanında olmayan yeni saldırılar için alarm verememesidir. Anormallik tabanlı STS'de ise saldırı olmayan aktiviteler için çeşitli bilgiler yardımı ile kullanıcı profilleri oluşturulur. Daha sonra bu profiller için çeşitli makine öğrenmesi ya da matematiksel modeller yardımı ile bir eşik değeri belirlenir. Yeni aktivite bu eşik değerini geçerse saldırı olarak nitelendirilir ve alarm tetiklenir. Bu yöntem imza tabanlı STS'ye göre yeni saldırıları tespit etmede daha başarılır ancak daha fazla yanlış alarm vermektedir.

Sürekli gelişen bilişim ürünleri ve değişen yaşam şartları nedeni ile uygulamalar genişlemekte veya çeşitlenmektedir. Bu değişim yeni sistem açıklıklarını buna bağlı olarak da yeni saldırı yöntemlerini beraberinde getirmektedir. Anormallik tabanlı STS'lerin yeni saldırı tiplerini tespit etmede imza tabanlı STS'lere göre daha başarılı olması ve makine öğrenmesi yöntemlerinin gün geçtikçe daha başarılı sonuçlar alması nedeni ile makine öğrenmesi tabanlı analiz yöntemlerini kullanan birçok STS modeli geliştirilmiştir. Mukkamala ve diğerleri DARPA veri seti yardımı ile analiz yöntemi olarak Destek Vektör Makineleri (DVM – Support Vector Machines) ve Yapay Sinir Ağları (YSA – Artificial Neural Networks) kullanan iki farklı STS modeli geliştirmişlerdir. Geliştirilen her iki model ile de %99'un üzerinde başarı oranı elde etmişlerdir [2]. Depren ve diğerleri KDDCup99 veri setini kullanarak denetimsiz makine öğrenmesi yöntemi olan Öz düzenleyici Haritalar (SOM – Self Organizing Map) ile eğitilen bir model geliştirmişlerdir. Geliştirilen bu model gelen saldırıları %99,1 oranında doğru tespit etmeyi başarırken, %1,32 oranında yanlış alarm vermiştir [3]. Depren ve diğerleri yapmış oldukları diğer bir çalışmada bir önceki çalışmada kullanılan aynı veri seti üzerinde J48 karar ağacı yöntemini kullanarak geliştirdikleri modelde, saldırı tespit oranını %99,9'a yükseltmeyi, yanlış alarm oranını da %1,25'e düşürmeyi başarmışlardır [4]. Sazlı ve Tanrıkulu yapmış oldukları çalışmada DARPA veri setini kullanan en iyi YSA modelini bulmak için tek katmanlı ve çok katmanlı olmak üzere birkaç farklı model geliştirmişlerdir. Analizlerin sonucunda tek katmanlı hiçbir YSA modelinde istenilen başarı oranı sağlanamaz iken, en iyi başarı oranı gizli katmanlarında sırası ile 10 ve 20 nöron kullanan model ile alınmıştır [5]. Tajbakhsh ve diğerleri KDDCup veri setini bulanık mantık birliktelik kuralları ile kümeledikleri modelde %91 oranında doğru tespit oranı elde etmişlerdir [6]. Wang ve diğerleri yapmış oldukları çalışmada önermiş oldukları bulanık kümelemeli yapay sinir ağları modelini (FC-YSA); karar ağaçları, Naive Bayes (NB) ve YSA ile karşılaştırmak için 5 sınıfı olan KDDCup veri setini kullanmışlarıdır. Önerilen bu yöntem ile normal, DoS ve PRB sınıflarının tespitinde diğer yöntemlere benzer sonuçlar alınırken; R2L ve U2L sınıflarının tespitinde kayda değer iyileşmeler sağlanmıştır [7]. Sağıroğlu ve diğerleri KDDCup ile oluşturulan 5 farklı test kümesini 3 farklı YSA modeli ile eğitmişler ve ortalama olarak %91,09 doğruluk oranı elde etmişlerdir [8]. Liu ve diğerleri SOM ağları ile geliştirdikleri sanal makineler üzerinde çalışan STS uygulamalarında % 98'e varan başarı oranı elde etmişlerdir [9]. Yıldız ve diğerleri bayesçi çoklu değişim noktası modeli ile VoIP ağlar için çalışan STS modeli geliştirmişlerdir. Saldırı ve trafik şiddetlerinin düşük ya da yüksek olması durumlarına göre analizler yapılmış, her iki şiddetin de yüksek olduğu durumda %90'lara varan doğru tespit oranı elde etmişlerdir [10]. Erhan ve diğerleri yapmış oldukları çalışmada K-SVD ile oluşturulan sözlüğü kullanarak eşleştirme algoritması yardımı ile DDOS atakları için eşik değeri tespit etmişlerdir [11]. Shakya ve Kaphle SOM ile YSA modellerini karşılaştırdıkları çalışmada SOM modellerinden daha iyi sonuçlar elde etmişlerdir [12]. Kaya ve diğerleri KDDCup veri setini kullanarak YSA, DVM, bayes ağları, karar ağaçları ve en yakın k komşu (k-nn) algoritmalarının performans ve başarı oranlarını karşılaştırmışlardır. Yapılan analizler sonucunda en iyi başarı oranının %99,9 olarak karar ağacı ile alındığını ve en hızlı algoritmanın 18 saniye ile bayes ağları olduğunu tespit etmişlerdir [13]. Kaya ve Yıldız yapmış oldukları

çalışmada literatürde var olan STS modellerini karşılaştırmak için SCI, E-SCI ve EBSCO'da taranan 65 farklı makaleyi incelemişlerdir. Bu incelemeler sonucunda en çok kullanılan veri setinin KDDCup99 olduğunu, en iyi başarı oranını YSA'nın verdiğini ancak bazı sınıfların tespitinde DVM'nin daha başarılı olduğunu belirtmişlerdir [14].

Sınıflama algoritmaları kendi aralarında kıyaslandığında bazen benzer hatalar yapsalar da, bazı durumlarda bir sınıfın yapısından kaynaklanan, sadece o sınıfa ait hatalar yapılabilmektedir. Bu özel durumların önüne geçerek hata oranlarını azaltmak için ortaya atılan topluluk (ensemble) yöntemlerde iki ya da daha fazla algoritma çeşitli matematiksel ya da istatiksel yöntemler ile birleştirilerek birlikte kullanılır. Mukkamala ve diğerleri topluluk yönteminin sınıflamaya etkisini göstermek için DARPA veri setini kullanan YSA, DVM ve çok değişken uyarlamalı regresyon düzlemleri (MARS - Multivariate Adaptive Regression Splines) yöntemleri ile oluşturmuş oldukları üç STS modelini, bu yöntemleri birleştirip kullandıkları topluluk STS modeli ile karşılaştırmışlardır. Yapılan analizler sonucunda topluluk yöntemi %99'un üzerinde başarı oranı ile en iyi skoru elde etmiştir [15]. Peddabachigar ve diğerleri geliştirdikleri DVM ve karar ağaçları yöntemlerini kullanan, beş sınıfı olan KDDCup veri seti ile geliştirdikleri topluluk model ile bu yöntemleri tek başına kullanan modelleri kıyaslamışlardır. Her bir sınıf türünde de topluluk modeli ile diğer yöntemlere göre benzer ya da daha iyi sonuçlar alınmıştır [16]. Sivatha Sindhu ve diğerleri C4.5 karar ağacı yöntemini modifiye ettikleri YSA ile birleştirerek %98,4 doğru saldırı tespiti yapmışlardır [17].

Sınıflama algoritmaları kadar kullanılan veri setindeki özniteliklerde bu algoritmaların başarısını etkileyen önemli unsurlardan biridir. Doğru özniteliklerin belirlenmesi için veri setinin daha küçük boyuta indirgendiği boyut düşürme ve öznitelik seçme yöntemleri sınıflama başarısını artırmak için birçok çalışmada kullanılmıştır. Horng ve diğerleri KDDCup99 veri setinden leave-one-out olarak adlandırdıkları öznitelik seçme yöntemi ile bazı öznitelikleri eleyerek oluşturulan yeni veri setini hiyerarşik kümeleme yöntemi ile kümeledikten sonra DVM ile sınıflayarak bir model geliştirmiş ve %95,72 doğruluk oranı elde etmişlerdir [18]. Amiri ve diğerleri karşılıklı enformasyon (MI - Mutual Information), ileriye doğru öznitelik seçimi (forward feature selection) ve benzer korelasyon tabanlı öznitelik seçim (near correlation-based feature selection) yöntemlerini kullanarak üç farklı veri seti elde etmişlerdir. Bu yeni veri setlerini en küçük kareler destek vektör makineleri (LSDVM - Least squares support vector machines) ile sınıflamışlar, MI ile oluşturulmuş veri setinde %90 doğru tespit oranı ile en iyi skoru elde etmişlerdir [19]. Li ve diğerleri önerdikleri öznitelik seçme yöntemi ile KDDCup99 veri setinden 19 kritik özniteliği seçerek oluşturdukları veri setini DVM ile sınıflayarak %98,62 başarı oranı elde etmişlerdir [20]. Ambusaidi ve diğerleri önerdikleri esnek Mİ yöntemini literatürde var olan MI yöntemi ile kıyaslamak için bu yöntemleri KDDCup, NSL-KDD ve kyoto2016 veri setlerine uygulayarak öznitelik seçimi yapmışlardır. Yeni veri setlerini çeşitli sınıflama algoritmaları ile analiz ederek esnek Mİ algoritmasında daha iyi sonuçlar alındığını belirtmişlerdir [21]. Zhu ve diğerleri çeşitli öznitelik seçimi ve boyut düşürme tekniklerini boyut düşürülmemiş veri ile kıyaslamak için literatürde var olan birkaç sınıflama ve kümeleme algoritması kullanarak farklı modeller geliştirmişlerdir. Analizlerin sonucunda I-NSGA-III öznitelik seçim yöntemini ve Hiyerarşik büyüyen SOM ağlarını birlikte kullanan modelin %99,24 doğru tespit oranı ile en iyi skoru elde ettiğini ve zamandan tasarruf sağladığını belirtmişlerdir [22]. Singh ve diğerleri Tekil Değer Ayrışımı (principal component analysis) ve genelleştirilmiş ayrım analizi (generalized discriminant analysis) yöntemleri ile boyutu düşürülmüş verileri; SOM ve C4.5 karar ağacı yöntemleri ile analiz ederek genelleştirilmiş ayrım analizinin her iki yöntem içinde daha iyi sonuçlar verdiğini belirtmişlerdir [23]. Varma vd. önerdikleri bulanık entropi tabanlı karınca koloni boyut düşürme yöntemini kullanarak iris veri setinden elde ettikleri yeni veri setini boyutu düşmemiş veri seti ile kıyaslamak için J48, rastgele ağaçlar, rastgele orman ve JRIP algoritmaları kullanan çeşitli modeller geliştirmişlerdir. Analizlerin sonucunda boyutu düşmemiş veri seti ile ortalama %99,58 doğruluk oranı elde ederken, önerilen yöntem ile elde edilen 13 özniteliği olan veri setinde %99,69 başarı oranı elde etmişlerdir [24].

Veri setlerindeki öznitelikler sınıflama performansını etkileyen en önemli unsurlardan biridir. Öznitelik sayısının az olması bazı durumlarda sınıfların düzgün olarak ayrışamamasına neden olmaktadır. Öznitelik sayısının fazla olması durumunda ise karşımıza eğitim zamanındaki artış, gürültüsü çok olan özniteliklerin doğruluk oranını düşürmesi gibi problemler çıkmaktadır. Bu nedenle özniteliklerin yeterli sayıda ve doğru olarak belirlenmesi gerekmektedir. Veri setlerindeki örnek sayılarının çokluğu düşünüldüğünde öznitelik belirleme işleminin insan eliyle yapılmasının neredeyse imkânsız bir hal alması bu problem için boyut düşürme ve öznitelik seçme algoritmalarının geliştirilmesine neden olmuştur. Boyut düşürme yöntemi veri setine çeşitli yöntemler uygulanarak daha küçük bir boyuta haritalanma işlemidir. Öznitelik seçim yöntemleri ise bazı özniteliklerin elenerek kalanların hiç değişime uğratılmadan kullanıldığı yöntemlerdir. Çalışmanın ana amacı öznitelik yöntemlerinin çeşitli sınıflayıcılar üzerindeki başarımını incelemek ve aynı zamanda aşırı öğrenme makinalarının çok sınıflı ve oldukça büyük veri ile analizi yapılan saldırı tespit sistemlerinde gerek hız gerekse başarı performansları açısından bilinen yöntemlere alternatif olarak kullanıp kullanılamayacağını belirlemektir.

Bu çalışmada 494021 veriye sahip KDDCup veri seti üzerinde Ki-Kare, Bilgi Kazancı, Kazanım Oranı, oneR, Gini Katsayısı, reliefF, genetik, geriye doğru ve ileriye doğru olmak üzere 9 tane öznitelik seçim algoritması uygulanarak elde edilen yeni veri setleri Aşırı Öğrenme Makineleri, destek vektör makineleri ve en yakın k komşu

yöntemleri ile sınıflandırılarak elde edilen analiz sonuçları boyutu düşürülmemiş veri seti ile kıyaslanmıştır. Çalışmanın ikinci bölümünde kullanılan öznitelik seçim ve sınıflama algoritmalarından bahsedilmiş, üçüncü bölümünde geliştirilen modeller ve elde edilen analiz sonuçları değerlendirilmiş ve son bölümünde çalışma özetlenerek sonlandırılmıştır.

2. YÖNTEMLER (METHODS)

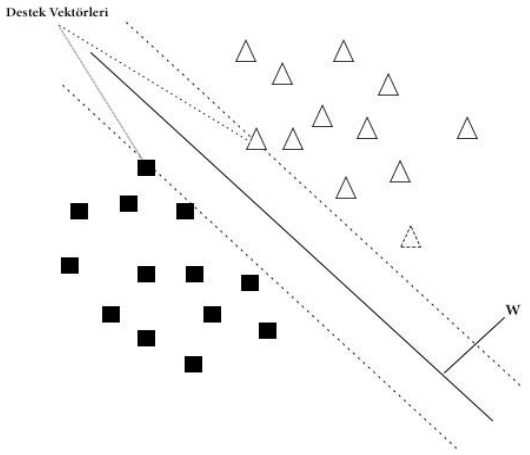
2.1. Sınıflama Algoritmaları (Classification Algorithms)

2.1.1. En Yakın K Komşu (K nearest Neigborhood)

En yakın k komşu (k-nn) algoritması, eğitim kümesinde tutulan verilere dayalı sınıflama yapan denetimli makine öğrenmesi yöntemlerinden biridir. Denetimli sınıflama algoritmalarının birçoğunda eğitim kümesi kullanılarak ön eğitim işlemleri ile bazı parametreler belirlenir ve test verileri, eğitim verilerine ihtiyaç duyulmadan belirlenen bu parametreler kullanılarak sınıflandırılır. K-nn algoritmasında ise ön eğitim işlemine ihtiyaç duyulmazken test verileri her seferinde eğitim kümesi kullanılarak sınıflandırılmaktadır. Bu nedenle k-nn algoritmasında ilk olarak etiketlenmiş veriler yardımı ile bir eğitim kümesi oluşturulur. Daha sonra k parametresi ve bir uzaklık fonksiyonu (minkowski, Öklid vb.) seçilir. Yeni bir veri ile karşılaşıldığında seçilen uzaklık algoritması kullanılarak bu verinin eğitim kümesindeki verilere olan uzaklığı tek tek hesaplanır. Daha sonra uzaklığı en küçük olan k adet veri eğitim kümesinden seçilerek sınıflama kümesi oluşturulur. Son adımda yeni verinin sınıfı, sınıflama kümesinin en çok içerdiği sınıf olarak belirlenir ve model sonlandırılır.

2.1.2. Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makinaları (DVM); doğru, düzlem ya da hiper düzlem yardımı ile verilerin iki sınıfa ayrıldığı bir makine öğrenmesi yöntemidir. Doğrusal olarak ayrışabilen veriler için sıkça kullanılan bu yöntem, çekirdek fonksiyonları yardımı ile verileri doğrusal olarak ayrıştırılabilir duruma getirebildiği için doğrusal olarak ayrışamayan veriler için de kullanılabilmektedir. Bu yöntemdeki asıl amaç, hataların karesini en aza indirecek ayracı belirlemektir [25]. Şekil 1'de görüldüğü gibi hatayı en az yapan ve birbirine paralel iki destek vektörü seçilerek, bu düzlemler arasındaki uzaklık maksimum yapılır. Daha sonra bu iki vektörün orta noktasındaki w vektörü seçilir. Son adımda ise yeni x için $y = w^t x + b$ işleminin sonucu hesaplanarak verinin sınıfına karar verilir. $y \leq 0$ durumu verinin birinci sınıfa, $y > 0$ durumu ise verinin ikinci sınıfa ait olduğunu temsil etmektedir.

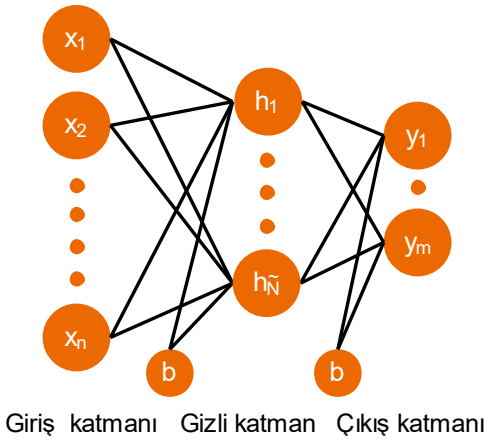


Şekil 1. Destek vektörleri (Support Vectors)

DVM diğer makine öğrenmesi tekniklerinden farklı olarak verileri sadece iki sınıfa ayırma işlemi yapmaktadır. Üç ya da daha fazla sınıfa sahip veri setlerinde, bire karşı bir (OVO, one versus one), ya da bire karşı hepsi (OVA, one versus all) ayrıştırma yöntemleri kullanılarak çok sınıflı verilerin sınıflandırılma işlemi DVM ile yapılabilir. Çalışmamızda OVA ayrıştırma tekniği kullanılmıştır. İlgili yöntemde farklı sınıf sayısı kadar ikili sınıflayıcılar oluşturulur. Veriler, ilgilenilen sınıf etiketi 1, diğer tüm sınıftaki veriler -1 olacak şekilde etiketlenerek sınıflandırma işlemi gerçekleştirilir. Bu işlem her bir sınıf için oluşturan modeller için ayrı ayrı tekrarlanır. Sınıflayıcı modellerden elde edilen sonuçlar birleştirilerek gerçek sınıf bilgisi elde edilir.

2.1.3. Aşırı Öğrenme Makineleri (Extreme Learning Machines)

Aşırı Öğrenme Makineleri (ELM) ilk olarak Huang ve arkadaşları tarafından 2006 yılında ortaya atılan, girdi katmanı, bir adet gizli katman ve çıktı katmanı olmak üzere üç katmandan oluşan tam bağlı bir yapay sinir ağı (YSA) modelidir [26]. Girdi katmanı verilerin okunduğu katmandır. Bu katmanda her bir nöron bir özelliği temsil ettiği için öznitelik sayısı kadar nöron bulunmaktadır. Çıktı katmanı ise sınıfların belirlendiği katmandır. Bu katman oluşturulan modele göre tek bir nöron içerebileceği gibi sınıf çeşidi kadar nöron da içerebilmektedir. Gizli katman ise girdi katmanı ile çıktı katmanı arasında yer alan ve verilerin ara işleme maruz kaldığı katmandır. Gizli katmandaki nöron sayısını belirleyen standart bir kural olmamasına karşın eğitimin kalitesini etkileyen önemli bir faktördür. n adet özniteliği ve m adet sınıfı olan bir veri seti için gizli katmanında $\tilde{N}$ adet nöron olan üç katmanlı bir YSA mimarisi şekil 2'de gösterilmiştir.



Şekil 1. Aşırı öğrenme makinesi modeli (Extreme Learning Machines Model)

Klasik yapay sinir ağlarında öğrenme süreci, geri yayılım algoritması kullanılarak, bilinen çıkış değerleriyle ağın üretmiş olduğu çıkış değerleri arasındaki hatayı sürekli azaltacak şekilde her iki katmandaki ağırlık değerlerini iteratif olarak güncelleyerek gerçekleştirilmektedir. Aşırı öğrenme makinalarında ise giriş katmanındaki ağırlıklar rastgele atanarak, gizli katmanla çıkış katmanı arasındaki ağırlıklar analitik bir denklem sistemi yardımıyla çok daha hızlı ve etkin bir şekilde belirlenebilmektedir. ELM, gradyent tabanlı öğrenme algoritmalarıyla eğitilen klasik YSA ağlarına oranla birçok avantaja sahiptir. Bu avantajlar aşağıdaki gibi sıralanabilir;

ELM'nin öğrenme süreci son derece hızlıdır. Bu süre genelde saniyeler seviyesinde, hatta bazı uygulamalarda saniyeden bile daha kısa olmaktadır.

ELM bir çok durumda türev tabanlı geri yayılım algoritmasından daha iyi genelleme yeteneğine sahiptir.

Klasik türev tabanlı eğitim algoritmaları ve diğer öğrenme algoritmaları yerel minimuma takılma, uygun olmayan öğrenme oranı, aşırı öğrenme ve ezberleme gibi birçok durumla karşı karşıya kalabilmektedir. Bu problemleri gidermek için erken durdurma, regülasyon parametreleri ekleme, ağrılık bozma ve geçerlilik seti kullanma gibi yöntemlere başvurulmaktadır. ELM öğrenme algoritması, bu tür ara işlemler olmaksızın doğrudan çözüme ulaşma eğilimindedir ve bu nedenle klasik yapay sinir ağlarında kullanılan öğrenme algoritmalarından daha basittir.

Geleneksel türev tabanlı öğrenme algoritmaları yalnızca türevi alınabilen aktivasyon fonksiyonlarını kullanabilirken ELM öğrenme algoritması kesikli ve türevi alınamayan birçok aktivasyon fonksiyonunu kullanabilmektedir [24].

Girdi değerleri, $x_i = [x_{i1}, x_{i2}, \dots x_{in}] \in R^N$, çıktı değerleri $t_i = [t_{i1}, t_{i2}, \dots t_{im}] \in R^M$ olmak üzere (x_i, t_i) değer çiftlerinden oluşan N adet örneğe sahip veri seti için, gizli katmanında $\tilde{N}$ adet düğüm bulunan ve aktivasyon fonksiyonu g(x) olarak verilen tek katmanlı yapay sinir ağı matematiksel olarak eşitlik 1'deki gibi modellenebilir.

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i . x_j + b_i) = o_j,\ j = 1, .., N \quad (1)$$

$w_i = [w_{i1}, w_{i2}, \dots w_{in}]^T$ gizli katmanda i'ninci düğüm ile giriş katmanını bağlayan ağırlıkları, $[\beta_{i1}, \beta_{i2}, \dots \beta_{im}]^T$ gizli katmandaki i'ninci düğümü çıkış katmanına bağlayan ağırlık değerlerini, b_i ilgili düğüm için eşik değerini, "." operatörü ise iki vektörün skaler çarpımını göstermektedir. Geri yayılım algoritmasıyla eğitilen tek katmanlı YSA'da olduğu gibi ağın eğitimindeki amaç, bilinen çıkış değerleri t_j ile ağın ürettiği çıkış değerleri o_j değerleri arasındaki farkı minimum yapmaktır. Hatayı sıfır yapmak üzere o_j ve t_j değerleri eşitlenirse aşağıda verilen eşitlik 2 elde edilir.

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i . x_j + b_i) = t_j, j = 1, \dots, N \quad (2)$$

Daha farklı bir şekilde eşitlik 3 gibi ifade etmek gerekirse;

$$H\beta = T$$

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 . x_1 + b_1) & \cdots & g(w_{\tilde{N}} . x_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(w_1 . x_N + b_1) & \cdots & g(w_{\tilde{N}} . x_N + b_{\tilde{N}}) \end{bmatrix}_{Nx\tilde{N}}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}xm} \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}xm} \quad (3)$$

Eşitlik 3'te verilen denklem sisteminin çözümü $\check{\beta} = H^{\dagger}T$ olarak elde edilir. $H^{\dagger}$ genelleştirilmiş ters Moore-Penrose matrisi olarak adlandırılır. Eğitim seti $N = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$aktivasyon fonksiyonu g(x) ve gizli nöron sayısı $\tilde{N}$ olan bir elm modelinin eğitimi kısaca 3 adımda özetlenebilmektedir [24]. Bu adımlar:

Her $i = 1, \dots, \tilde{N}$değeri için giriş katmanı ile gizli katman arasındaki ağırlık vektörleri (w_i) ve (b_i) bias değerleri rastgele belirlenir.

Gizli katman çıkış matrisi H hesaplanır ardından $\check{\beta} = H^{\dagger}T$ eşitliği yardımıyla $\check{\beta}$ değerleri hesaplanır.

2.2. Öznitelik Seçim Yöntemleri (Feature Selection Methods)

2.2.1. Ki Kare (Chi Square)

Randy Kerber tarafından 1992 yılında ortaya atılan ve Huan lui ve Rudy Setiono tarafından 1995 yılında geliştirilen X^2 testi olarak da bilinen Ki-kare yöntemi değişkenlerin veri setini tanımlamaya uygun olup olmadığını belirlemek içinde kullanılabilir [27]. Ki-kare

testinde H_0 ve H_1 olmak üzere iki hipotez bulunmaktadır. H_0 veri setindeki değişkenlerin uygun olduğu, H_1 ise veri setindeki değişkenlerin uygun olmadığı hipotezidir. İki aşaması olan bu testin ilk aşamasında gözlenen değerlerin gerçek sınıflara göre ki-kare(X^2) istatistiği hesaplanır. X^2 değeri sıfır ile pozitif sonsuz arasında değerler alabilir. Bu değerin sıfıra yaklaşması gözlenen frekans değerleri ile beklenen frekans değerlerinin daha uyumlu olduğunu gösterir. Bu değerin çok büyük olması uyumsuzluğu işaret etmektedir. Bu nedenle testin ikinci aşamasında ilk aşamada hesaplanan X^2 değeri Ki-kare dağılımındaki belirlenen eşik değeri ile kıyaslanır. Bu eşik değeri önemlilik seviyesine ve serbestlik derecesine bakılarak bulunmaktadır. Önemlilik seviyesi testi yapan kişi tarafından belirlenen yüzde değeridir, serbestlik derecesi ise veri setindeki öznitelik sayısının bir eksiğidir. Hesaplanan değer belirlenen değerden büyük ise H_1 hipotezi, küçük ise H_0 hipotezi kabul edilir. Eşitlik 4'de Ki-kare istatistiğinin nasıl hesaplandığı formülüze edilmiştir.

$$X^2 = \sum_{i=1}^{n} \frac{(o_i - e_i)^2}{e_i} \quad (4)$$

Bu eşitlikte n veri setindeki öznitelik sayısını, o_i i'inci öznitelik için gözlenen frekans değerini, e_i ise i'inci öznitelik için beklenen frekans değerini temsil etmektedir.

2.2.2. Bilgi Kazancı (Information Gain)

Bilgi kazancı, veri seti özniteliklere göre bölündüğünde tahmini kaybı hesaplamada kullanılan entropiye dayalı yöntemlerden biridir. Entropi sistemin düzensizliğini ya da belirsizliğini belirleyen 0 ile 1 arasında bir değerdir. Entropi değerinin 1'e yaklaşması sistemin daha çok bilgi içerdiğini göstermektedir. Bilgi kazancı yönteminin ilk aşamasında, verilen bir veri setinin sınıf etiketleri için entropi değeri eşitlik 5'de formülüze edildiği gibi hesaplanır.

$$E = -\sum_{i=1}^{n} \left(log_2 \frac{ns(i)}{N}\right) * \frac{ns(i)}{N} \quad (5)$$

Bu eşitlikte n sınıf sayısını, ns(i) i'inci sınıf için örnek sayısını, N ise toplam örnek sayısını temsil etmektedir.

Bu yöntemin ikinci aşamasında veri setindeki her bir öznitelik için entropi değeri hesaplanır ve bu yeni entropi değeri ilk aşamada bulunmuş olan değerden çıkarılarak bilgi kazancı hesaplanır. Bilgi kazancı veri setinin bölünme sonrası temsil değerini göstermektedir. Bu nedenle bu değerin büyük olması beklenmektedir. Bilgi kazancı yöntemi ile öznitelik seçimi yapılırken sistemi tanımlamada yetersiz kalan değişkenler veri setinden çıkarılır ve kalan değişkenler sistemi eğitmek için kullanılır. Eşitlik 6'da her bir öznitelik için entropi değerinin, eşitlik 7'de ise bilgi kazancı değerinin hesaplanması formülüze edilmiştir.

$$E(i) = \sum_{k=1}^{n} \frac{ns(k)}{N} * \sum_{m=1}^{nc} -\frac{nsc(k,m)}{ns(k)} \left(log_2 \frac{nsc(i,m)}{ns(i)}\right) \quad (6)$$

$$B(i) = E(i) - E \quad (7)$$

Eşitliklerde E(i) i'inci öznitelik için entropi değerini, n i'inci özniteliğin alabileceği farklı değer sayısını, ns(k) i'inci özniteliğin k değerine ait örnek sayısını, N veri setindeki toplam örnek sayısını, nc veri setindeki sınıf sayısını, nsc(k,m) i'inci özniteliğin k değerine ait m sınıfını temsil eden örnek sayısını, B(i) bilgi kazancını, E ise eşitlik 5'te hesaplanan entropi değerini temsil etmektedir.

2.2.3. Kazanım Oranı (Gain Ratio)

Kazanım oranı, bilgi kazancı yöntemine alternatif olarak aynı amaç doğrultusunda kullanılan öznitelik seçim yöntemlerinden biridir. Veri setinde bir öznitelik çok fazla farklı değere sahip olduğunda o öznitelik için her bir farklı değere düşen örnek sayısı düşük olmaktadır. Bu nedenle o öznitelik için hesaplanan entropi değeri küçük, bilgi kazancı ise büyük çıkmaktadır. Bilgi kazancı yönteminde de anlatıldığı gibi bu değerin büyük çıkması o değişkenin veri setini tanımlamada iyi olduğunu göstermektedir. Özniteliğin alabileceği farklı değer sayısının çok olması durumunda bilgi kazancı yönteminin o özniteliği iyi bir seçici olarak seçmesi sistemin ezberleme yapmasına neden olabilmektedir. Kazanım oranı bu probleme çözüm olarak her bir öznitelik için bölünme bilgisini hesaplar ve elde edilen bilgi kazancını bölünme bilgisine bölerek kazanım oranını hesaplar. Bu oranın 1'e yaklaşması o değişkenin veri setini tanımlamada başarılı olduğunu göstermektedir. Eşitlik 8'de bölünme bilgisi, eşitlik 9'da ise kazanım oranı verilmiştir.

$$S(i) = -\sum_{k=1}^{n} \frac{ns(k)}{N} * \left(log_2 \frac{ns(k)}{N}\right) \quad (8)$$

$$K(i) = \frac{B(i)}{S(i)} \quad (9)$$

Bu eşitliklerde S(i) i'inci öznitelik için bölünme bilgisini, n i'inci özniteliğe ait farklı değer sayısını, ns(k) i'inci özniteliğin k değerine ait örnek sayısını, N toplam örnek sayısını, K(i) i'inci öznitelik için kazanım oranını B(i) ise eşitlik 7'de hesaplanan bilgi kazancı değerini temsil etmektedir.

2.2.4. Gini Katsayısı (Gini Index)

Gini katsayısı, kazanım oranı ve bilgi kazancı yöntemlerine alternatif olarak geliştirilmiş diğer bir öznitelik seçim yöntemidir. Bu yöntem diğer iki yöntemde de olduğu gibi her bir öznitelik için kazanım hesaplayarak sıralama yapmaktadır. Bu yöntem diğer iki yöntemden farklı olarak entropi değerini kullanmamaktır. Gini yönteminin ilk aşamasında kullanılan veri setinin etiket değeri ve her bir öznitelik için gini katsayısı hesaplanır. Ardından her bir öznitelik için ayrı ayrı gini kazanım değeri o öznitelik için hesaplanan gini katsayısının etiketler

için hesaplanan gini katsayısından çıkarılması ile hesaplanır. Son aşamada kazanım oranı belirlenen eşik değerinin altında kalan öznitelikler veri setinden çıkarılır ve yeni veri seti oluşturulur. Eşitlik 10'da etiket değeri için, Eşitlik 11'da ise her bir öznitelik için gini katsayısının hesaplanması formülüze edilmiştir.

$$Gini = \prod_{i=1}^{n} p(sınıf = i) \quad (10)$$

$$\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{p(değer = i)} \times \prod_{\mathrm{j=1}}^{\mathrm{m}} \frac{\mathrm{N(değer=i \,\&\, } sınıf = j\mathrm{)}}{\mathrm{N(değer=i)}} \quad (11)$$

Eşitlik 10'da n sınıf sayısını, eşitlik 11'de n, k değişkeni için farklı değer sayısını m ise sınıf sayısını temsil etmektedir.

Bahsedilmiş olan bilgi kazancı, kazanım oranı ve gini katsayısı yöntemlerinde özellikler kategorik olarak düşünülmüştür. Devamlı değerler ise iki şekilde kategorik olarak yeniden düzenlenebilir. İlk yöntemde rastgele sayıda rastgele değerler belirlenir ve o değerlere göre özellik bölünür. İkinci yöntemde ise özellik sıralanır ve kaça bölüneceği seçilir. Ardından her bir bölünme noktası için kazanımlar hesaplanır ve en iyi olanı seçilir.

2.2.5. OneR

OneR kuralı değişkenleri hata oranlarına göre sıralayan bir öznitelik seçimi yöntemidir. Diğer yöntemlere göre daha basit olan bu yöntemdeki amaç her bir değişkenin tek başına kullanılması durumunda oluşacak hata değerini hesaplayarak değişkenleri sıralamaktır. Bu doğrultuda veri setindeki her bir değişken için bir model oluşturulur. Oluşturulan her bir model için veri seti modeli oluşturan değişkenin alabileceği farklı değerlere göre bölünür. Daha sonra her bir farklı değer için daha sık geçen sınıf temel sınıf olarak kabul edilir ve diğer sınıfa ait örnekler hata olarak kabul edilir. Her bir farklı değer için elde edilen hata toplandığı zaman modelin toplam hatası elde edilmiş olur. Modeller toplam hatası küçük olandan büyük olana doğru sıralandıktan sonra belirlenen eşik değerinin altında kalan modeli oluşturan değişken veri setinden çıkartılarak yeni veri seti elde edilir. Birçok yöntemle karşılaştırıldığında daha basit olmasına rağmen bu yöntemin birçok veri seti için başarılı sonuçlar verdiği görülmüştür.

2.2.6. ReliefF

Kira ve arkadaşları tarafından 1992 yılında geliştirilen Relief algoritması öznitelikleri aralarındaki ilişkiye göre ağırlıklandıran bir öznitelik seçim yöntemidir [28] Bu yöntem iki sınıfı olan veri setleri için başarılı sonuç verse de çoklu sınıfa sahip veri setleri için çalışmamaktadır. Bu problemi çözmek için 1994 yılında Kononenko çoklu sınıfı olan veri setleri içinde çalışan ReliefF algoritmasını geliştirmiştir [29]. Bu yöntemin ilk aşamasında tüm özniteliklerin ağırlıkları 0 olarak belirlenir. Daha sonra her bir adımda veri setinden rastgele bir veriyi seçer ve bu veri ile aynı sınıfa ait en yakın k (k değeri sınıf sayısının bir eksiğidir) adet veri bulur ardından her bir farklı sınıfa ait en yakın veriler bulunur. Sonrasında her bir özelliğe ait ağırlıklar bu veriler kullanılarak güncellenir. Son aşamada belirlenen koşulu sağlamayan özellikler veri setinden atılarak yeni veri seti oluşturulur. Eşitlik 12'de reliefF algoritması, Eşitlik 13'de ikili değerler için uzaklık hesaplanması, Eşitlik 14'de ise devamlı değerler için uzaklık hesaplanması formülüze edilmiştir.

$$W(x^a) = W(x^a) - \frac{\sum_{j=1}^{k} uzaklık(A,R_i,H_j)}{m \times k} + \frac{\sum_{C \neq sınıf(R_i)} \left[\frac{P(C)}{1-P(sınıf(R_i))} \times \sum_{j=1}^{k} uzaklık(A,R_i,M_j) \right]}{m \times k} \quad (12)$$

$$uzaklık(A, I_1, I_2) = \begin{cases} 0, & I_1 = I_2 \\ 1, & I_1 \neq I_2 \end{cases} \quad (13)$$

$$uzaklık(A, I_1, I_2) = |I_1 - I_2| \times \frac{1}{\max(A) - \min(A)} \quad (14)$$

Bu eşitliklerde $W(x^a)$ a'ıncı özniteliğin ağırlığını, k sınıf sayısının bir eksiğini, m döngü sayısını, R_i i'inci döngüde seçilmiş olan veriyi, H_j seçilen veri ile aynı sınıfa ait j'inci yakın veriyi, M_j j'inci sınıfa ait seçilen veriye en yakın olan veriyi temsil etmektedir.

2.2.7. İleriye Doğru Öznitelik Seçimi (Forward Feature Selection)

İleriye doğru öznitelik seçimi (FFS); eğitim için en uygun öznitelik setini bulmayı hedefleyen bir öznitelik seçim yöntemidir. Öznitelik seçme işlemi azdan çoğa doğru yapıldığı için ileriye doğru olarak adlandırılmıştır. Yöntemde ilk olarak boş bir öznitelik seti ile başlanır. Daha sonra hedef hipotezi H_0 ve bunun karşıt hipotezi H_1 belirlenir. H_0; başarı oranını artırmak, doğru tespit oranı artırmak gibi isteğe bağlı olarak herhangi bir performans unsuru olabilmektedir. Ardından D özniteliği olan bir veri setinden bir öznitelik seçilerek oluşturulan sete eklenir ve eğitim yapılır. Eğer H_0 hipotezi sağlanıyorsa eğitim setine seçilmiş olan öznitelik eklenerek, H_1 hipotezi sağlanıyorsa oluşturulan eğitim seti değiştirilmeden aynı işlemler diğer öznitelikler içinde yapılır. D adet özniteliğin hepsi için aynı işlem tekrarlanarak yöntem sonlandırılır ve en uygun sonucu veren veri seti tespit edilmeye çalışılır.

2.2.8. Geriye Doğru Öznitelik Seçimi (Backward Feature Selection)

Geriye doğru öznitelik seçimi (BFS); FFS ile aynı amaç doğrultusunda kullanılan bir yöntemdir. FFS'den farklı olarak öznitelik eleme işlemi geriye doğru yani çoktan aza olacak şekilde yapılmaktadır. Bu nedenle yöntemde ilk olarak D adet özniteliği olan veri setinin tümü öznitelik seti olarak belirlenir. Daha sonra FFS'ye benzer şekilde H_0 ve H_1 hipotezleri belirlenir. Üçüncü adımda ise belirlenen öznitelik setinden bir öznitelik çıkarılarak eğitim işlemi yapılır. H_0 hipotezi sağlanırsa o öznitelik oluşturulan setten çıkarılarak, H_1 sağlanıyorsa öznitelik setinde

değiştirilmeden işlemler tüm öznitelikler için tekrar edilir ve model sonlandırılır. FFS ve BFS yöntemlerinin ikisi de en uygun öznitelik setini bulmayı hedefler ancak bulunan öznitelik setinin en uygun set olduğunu garanti edemezler.

2.2.9. Genetik Algoritma (Genetic Algorithm)

Genetik algoritma doğal seleksiyon, çaprazlama ve mutasyon tabanlı, biyolojiden ilham alan global arama optimizasyon tekniğidir. Bu yöntemde öncelikle aday çözümlerden oluşan bir popülasyon üretilir ve bu popülasyon belirlenen durdurma kriteri sağlanıncaya kadar seleksiyon, çaprazlama ve mutasyon adı verilen genetik işlemler aracılığı ile güncellenir. GA daha iyi çözümleri bulma sürecinde en iyi olanın hayatta kalması fikrini kullanır. GA tek bir çözümü kademeli olarak değiştirmektense bir çözüm popülasyonunu güncelleyerek arama yapması yönüyle geleneksel doğrusal olmayan optimizasyon tekniklerinden ayrılır. Klasik optimizasyon algoritmaları iterasyon noktalarının yerel özellikleri ile ilgilendiği için kolayca yerel ekstremum noktalarına takılabilirler. Bunun aksine GA sistematik aramaya ek olarak rasgele arama operatörü de kullandığından dolayı yerel minimum veya maksimum noktasına takılması önlenmiş olur.

GA, optimize edilecek parametrelerin bir dizi çözümüyle başlar. Çözümü oluşturan kromozomların her bir parametresi gen olarak adlandırılır ve parametreler ikili bit dizisi, tam ya da reel sayı şeklinde kodlanabilirler. Herhangi bir ön bilgi olmadığında ilk popülasyondaki her kromozom, düzgün dağılım kullanarak rastgele oluşturulur. Daha sonra her bir çözümün uygunluğunu belirlenen bir fonksiyon yardımı ile bulunurak büyükten küçüğe doğru bir sıralama yapılır. Bu sıralanan nesillerin yardımı ile mutasyon, çaprazlama gibi teknikler kullanılarak yeni nesiller üretilir ve istenilen başarı oranı elde edilene kadar bu işlemler yeni nesiller üzerinde de tekrar edilir. Bu özelliği ile genetik algoritma ile, bir çok algoritmaya göre daha yavaş çalışmasına rağmen daha yüksek başarı oranı elde edilir.

3. UYGULAMA (APPLICATION)

KDD veri seti Amerikan Hava Kuvvetleri ağına benzer yapıda tasarlanmış bir benzetim veri setidir [30]. Saldırı tespit çalışmalarında çokça tercih edilen bu veri setinde basit(basic), içerik(content), trafik(traffic) ve host adı altında 4 gruptan 41 özellik ve toplamda yaklaşık 4,8 milyon kayıt bulunmaktadır. Çalışmamızda kullanılan KddCup99 verilerinin %10'luk kısmında 4 ana türden (DOS (denial-of-service), R2L (uzak bir sunucudan gelen izinsiz erişim), U2R (yerel makineye olan izinsiz erişim), probing (gözetim ve diğer algılama saldırıları)) üretilmiş 22 saldırı ve normal kayıtlardan oluşan toplam 23 sınıf ve 494,021 kayıt vardır [30]. Bu kayıtlardan 97278 tanesi saldırı olmayan kayıtları, geriye kalan 396743 tanesi ise bahsedilen 22 saldırı tipinden birini temsil etmektedir. Veri setini daha anlamlı hale getirmek için ilk olarak bu veri setindeki sözel öznitelikler (servis, protokol tipi vb.) sayısal verilere dönüştürülmüş ardından her bir öznitelik için normalizasyon işlemi python kütüphaneleri kullanılarak yeni veri seti elde edilmiştir.

Uygulamamızın tüm analizleri Matlab ortamında yapılmıştır. Uygulamada ilk olarak anlatılan veri seti eğitim ve test olmak üzere iki alt veri setine parçalanmıştır. Deneyler 10 kat çapraz doğrulama yöntemi kullanılarak yapılmıştır. Daha sonra her eğitim veri seti ile ELM, DVM ve K-nn algoritmaları kullanılarak üç adet model oluşturulmuş ve her üç model için de test başarı oranları elde edilmiştir. Ardından eğitim veri seti üzerinde ki kare, bilgi kazancı (BK), kazanım oranı (KO), gini katsayısı (GK), oneR ve reliefF algoritmaları ile hesaplanan önem değerine göre, tüm veri setindeki öz nitelikler sıralanarak 6 farklı veri seti oluşturulmuştur. 6 farklı öznitelik seçim yönteminden elde edilen önem değerleri için bir eşik değeri belirleyerek bu eşik değerinin üzerinde önem derecesine sahip öz nitelikler seçilebilmektedir. Ancak eşik değerinin ne olacağını bilgisi açık değildir. Bu nedenle oluşturulan 6 veri setinin her biri için 1 öznitelik ile 40 öznitelik arasından en iyi test başarı oranını veren boyut ELM, K-nn ve DVM algoritmaları için ayrı ayrı belirlenmiştir. Uygulamanın son adımında her üç sınıflama yöntemi içinde FFS, BFS ve genetik algoritma uygulanarak toplamda 30 adet model oluşturulmuştur. Kullanılan veri setlerinin boyutu (B), başarı oranı (Accuracy - Acc), hassasiyet (Sensivity - Sens), yanlış alarm oranı (False Discovery Rate - FDR), f-ölçüt (F) değerleri [31] ve saniye bazında çalışma zamanı (Running Time – RT); DVM kullanan modeller için tablo 1'de, K-nn kullanan modeller için tablo 2'de, ELM kullanan modeller için tablo 3'te gösterilmiştir. Tabloda verilen performans ölçütleri on kat çapraz doğrulama sonucunda elde edilen ortalama değerlerdir. Deneyler esnasında her bir kat için bir çekirdek kullanıldığından dolayı çalışma süresi en yavaş veri setinin çalışma süresini göstermektedir.

Tablo 1. Destek Vektör Makineleri Analiz Sonuçları
(Support Vector Machines Analysis Results)

Kullanılan Yöntem	*B*	*Acc*	*Sens*	*FDR*	*F*	*RT*
----	41	0,970	0,353	0,036	0,741	64043s
Ki-Kare	16	0,971	0,391	0,032	0,805	49702s
BK	17	0,969	0,397	0,033	0,806	50361s
KO	21	0,998	0,582	0,067	0,957	46913s
GK	11	0,991	0,610	0,124	0,851	21187s
oneR	8	0,980	0,373	0,032	0,807	50781s
reliefF	8	0,974	0,270	0,083	0,691	35694s
FFS	20	0,997	0,710	0,059	0,903	44706s
BFS	32	0,997	0,636	0,067	0,959	54982s
Genetik	14	0,991	0,508	0,012	0,860	35950s

Tablo 2. K-nn Analiz Sonuçları
(K-nn Analysis Results)

Kullanılan Yöntem	*B*	*Acc*	*Sens*	*FDR*	*F*	*RT*
----	41	0,998	0,690	0,092	0,921	549,8s
Ki-Kare	6	0,998	0,694	0,039	0,923	26,5s
BK	15	0,998	0,702	0,107	0,922	211,2s
KO	21	0,999	0,718	0,023	0,948	277,3s
GK	17	0,998	0,702	0,100	0,926	242,2s
oneR	17	0,998	0,702	0,100	0,926	224,8s
reliefF	29	0,998	0,690	0,092	0,921	380,5s
FFS	20	0,999	0,727	0,020	0,956	265,4s
BFS	35	0,999	0,726	0,024	0,954	454,6s
Genetik	14	0,998	0,676	0,061	0,900	204,1s

Tablo 3. ELM Analiz Sonuçları
(ELM Analysis Results)

Kullanılan Yöntem	*B*	*Acc*	*Sens*	*FDR*	*F*	*RT*
----	41	0,974	0,170	0,041	0,944	2,79s
Ki-Kare	14	0,974	0,174	0,156	0,884	2,28s
BK	34	0,973	0,169	0,042	0,943	2,41s
KO	35	0,978	0,153	0,066	0,707	2,60s
GK	31	0,979	0,135	0,274	0,981	2,40s
oneR	3	0,979	0,192	0,298	0,836	2,24s
reliefF	29	0,972	0,180	0,187	0,787	2,37s
FFS	13	0,972	0,180	0,187	0,787	2,32s
BFS	41	0,972	0,134	0,276	0,974	2,78s
Genetik	14	0,975	0,216	0,165	0,743	2,36s

Çalışmada kullanılan öznitelik seçim algoritmaları iki kümede toplanabilmektedir. İlk kümeyi öznitelikleri önem sırasına göre sıralayan Ki-Kare, BK, KO, GK, oneR ve reliefF yöntemleri, ikinci kümeyi ise uygun öznitelikleri sınıflandırma performansına göre seçen FFS, BFS ve Genetik algoritma yöntemleri oluşturmaktadır. İlk kümede bahsedilen 6 yöntemde her bir öznitelik için sınıftan bağımsız olarak bir önem değeri hesaplanmakta ve daha sonra bu öznitelikler bu önem değerine göre yeniden sıralanmaktadır. İkinci kümedeki genetik algoritma yöntemi için ise aç gözlü tekniği kullanılarak öznitelikler bir defaya mahsus seçilmiş ve daha sonra her bir yöntem için bu seçilen öznitelikler kullanılmıştır. Geriye kalan BFS ve FFS yöntemleri ise her bir sınıflayıcı için ayrı ayrı uygulanarak farklı öznitelikler seçilmiştir. İlk kümede yer alan her bir yöntem kullanılarak önem değerine göre yeniden sıralanarak elde edilen yeni veri setlerindeki öznitelik sıralamaları ve Genetik algoritma kullanılarak seçilen öznitelikler tablo 4'te özetlenmektedir. Tablo 5'te ise FFS ve BFS algoritmaları kullanılarak seçilen öznitelikler her bir sınıflama yöntemi için gösterilmiştir.

Tablo 4. Özniteliklerin Yönteme Bağlı Önem Sırası
(Feature Ranking Order Depends on Methods)

Kullanılan Yöntem	*Öznitelik Sıralaması*
Ki-Kare	4, 5, 2, 3, 22, 34, 7, 29, 9, 37, 32, 35, 23, 24, 36, 33, 28, 39, 1, 38, 26, 12, 25, 6, 40, 31, 10, 11, 15, 30, 13, 27, 0, 17, 21, 16, 18, 19, 8, 14, 20
BK	4, 22, 2, 23, 35, 1, 32, 34, 33, 29, 28, 3, 5, 37, 24, 38, 25, 11, 31, 36, 30, 39, 40, 26, 27, 0, 9, 12, 7, 21, 15, 18, 16, 10, 13, 6, 17, 19, 14, 8, 20
KO	7, 12, 6, 1, 10, 3, 9, 25, 24 2, 11, 29, 38, 37, 35, 13, 4, 28, 5, 34, 36, 32, 22, 33, 23, 21, 31, 39, 26, 30, 40, 27, 0, 16, 17, 15, 18, 19, 8, 14, 20
GK	4, 2, 12, 6, 1, 22, 10, 3, 32, 9, 25, 7, 24, 11, 29, 38, 37, 35, 13, 28, 5, 34, 36, 33, 23, 21, 31, 39, 26, 30, 40, 27, 0, 16, 17, 15, 18, 19, 8, 14, 20
oneR	4, 22, 2, 35, 32, 34, 33, 1, 29, 28, 3, 37, 38, 24, 25, 5, 11, 31, 36, 30, 40, 39, 26, 27, 0, 9, 12, 7, 15, 18, 21, 16, 17, 10, 6, 13, 14, 19, 8, 20
reliefF	1, 25, 24, 37, 35, 22, 2, 23, 38, 32, 31, 3, 33, 11, 28, 39, 40, 26, 27, 30, 34, 29, 36, 7, 9, 21, 0, 4, 5, 13, 6, 10, 18, 12, 17, 16, 14, 19, 20, 8, 15
Genetik	1, 2, 4, 5, 6, 7, 13, 21, 22, 29, 31, 35, 37, 29

Tablo 5. FFS ve BFS Algoritmaları Kullanılarak Her Bir Sınıflayıcı İçin Seçilen Öznitelikler
(Selected Feature for Each Classifier by Using FFS and BFS)

Kullanılan Yöntem	*Öznitelik Sıralaması*
ELM + FFS	1, 2, 3, 4, 5, 6, 8, 9, 15, 16, 18, 20, 24
ELM + BFS	1, 2, 3, 4, 5, 6, 7, 8, 9,10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41
DVM+ FFS	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 20, 21, 28
DVM + BFS	1, 2, 3, 4, 5, 6, 7, 8, 9,10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 27, 29, 30, 33, 34, 36, 37, 38, 39
K-nn + FFS	1, 3, 4, 5, 6, 7, 8, 9, 12, 14, 16, 17, 18, 19, 20, 21, 24, 28, 31, 36
K-nn + BFS	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 24, 26, 28, 29, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 41

4. SONUÇLAR (CONCLUSIONS)

Bu çalışmada KDDCup99 veri seti üzerinde çeşitli öznitelik seçim algoritmaları kullanılarak çeşitli STS modelleri geliştirilmiştir. Ki-Kare, Bilgi Kazancı, Kazanım Oranı, Gini Katsayısı, OneR, ReliefF, Genetik, FFS ve BFS olmak üzere 9 farklı öznitelik seçim yöntemi kullanılarak elde edilen yeni veri setlerini orijinal boyuttaki veri seti ile karşılaştırılmak için DVM, K-nn ve ELM sınıflama algoritmaları kullanılmıştır. Sonuçlar incelendiğinde sınıflama algoritması olarak K-nn kullanan modellerin; ELM ve ya DVM kullanan modellere göre biraz daha iyi başarı oranları elde ettiği görülmüştür. Zaman bazında incelediğinde ELM kullanan modellerin en hızlı çalıştığı, DVM kullanan modellerin ise diğer iki modele göre çok yavaş çalıştığı gözlemlenmiştir. ELM nin hızlı çalışmanın nedeni geri yayılım algoritması yerine giriş katmanındaki ağırları rastgele atayarak gizli katmandaki ağırlıkları basit bir matris bölme işlemiyle belirlemesinden kaynaklanmaktadır. DVM algoritmasının yavaş çalışmasının nedeni, veri setinin 23 sınıf içermesinden dolayı bire karşı hepsi yönteminin yapısı gereği her sınıf için DVM algoritmasının 23 kez çalıştırılmasından kaynaklanmaktadır. Buna ek olarak DVM kullanan modellerin diğer iki sınıflama yöntemini kullanan modellere nazaran biraz daha fazla yanlış alarm verdiği görülmektedir. Modeller kendi aralarında kıyas edildiğinde en çok yanlış alarmı ELM ve oneR yöntemlerini birlikte kullanan modelin verdiği, en çok doğru tespitin K-nn ve FFS yöntemlerini birlikte kullanan model ile yapıldığı görülmektedir. Zaman bazında incelediğinde en hızlı 2,24s ile ELM ve oneR yöntemlerini birlikte kullanan modelin, en yavaş ise 64043s ile DVM modelini tek başına kullanan modelin çalıştığı gözlemlenmiştir. Öznitelik yöntemleri boyut bazında incelediğinde; En az boyuta K-nn sınıflama algoritması kullanan modeller için Ki-Kare yöntemi, ELM sınıflama algoritması kullanan modeller için oneR yöntemi, DVM kullanan modeller içinde reliefF ve oneR yöntemleri ile düşüldüğü görülmüştür. Her üç sınıflama yöntemi içinde BFS en yüksek boyutta kalan yöntem olmuştur. Yapılan analizler doğrultusunda öznitelik seçim algoritmaları ile boyutu düşürülmüş veri setleri ile orijinal boyuttaki veri setine göre daha iyi ve hızlı sonuçlar alındığı anlaşılmıştır. Sınıflama algoritmalarında ise DVM algoritması hem zaman bazında hem de başarı ölçütleri bazında diğer iki algoritmadan daha kötü sonuçlar almıştır. Bu nedenle tasarlanacak bir STS için öznitelik seçim algoritmalarının DVM dışında diğer iki algoritma ile kullanılması uygun görülmüştür. Aşırı öğrenme makinalarının özellikle hızlı olması, başarı oranlarının yüksek olması, eğitim algoritmasının basitliği nedeniyle kolayca gerçekleştirilebilir olması gibi nedenlerden ötürü özellikle online STS'lerde alternatif bir sınıflayıcı olarak kullanılabilir olduğu çıkarımı çalışmanın sonuçlarından bakılarak yapılmıştır.

KAYNAKLAR (REFERENCES)

[1] Özgür, A, Erdem, H. "Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması". *Bilişim Teknolojileri Dergisi*, 5(2), 41-48, 2012.

[2] S. Mukkamala, G. Janoski, ve A. Sung, "Intrusion detection using neural networks and support vector machines", **2002 International Joint Conference on Neural Networks (IJCNN '02)**, Honolulu, A.B.D., 1702–1707, 12-17 Mayıs 2002.

[3] M. O. Depren, M. Topallar, E. Anarim, ve K. Ciliz, "Network-based anomaly intrusion detection system using SOMs", IEEE **12th Signal Processing and Communications Applications Conference**, Kuşadası, Türkiye, 76–79, 30 Nisan 2004.

[4] O. Depren, M. Topallar, E. Anarim, ve M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", *Expert Syst. Appl.*, 29(4), 713–722, 2005.

[5] M. H. Sazlı ve H. Tanrıkulu, "Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması", **XII. Türkiye'de İnternet Konferansı**, Ankara, Türkiye, 8-10 Kasım 2007

[6] A. Tajbakhsh, M. Rahmati, ve A. Mirzaei, "Intrusion detection using fuzzy association rules", *Appl. Soft Comput.*, 9(2), 462–469, 2009.

[7] G. Wang, J. Hao, J. Ma, ve L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", *Expert Syst. Appl.*, 37(9), 6225–6232, 2010.

[8] Ş. Sağiroğlu, E. N. Yolaçan, ve U. Yavanoğlu, "Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi", *Gazi Üniversitesi Mühendis.-Mimar. Fakültesi Derg.*, 26(2), 325-340, 2011.

[9] J. Liu, S. Chen, Z. Zhou, ve T. Wu, "An Anomaly Detection Algorithm of Cloud Platform Based on Self-Organizing Maps", *Math. Probl. Eng.*, 2016, 2016.

[10] Ç. Yıldız, T. Y. Ceritli, B. Kurt, B. Sankur, ve A. T. Cemgil, "Attack detection in VOIP networks using Bayesian multiple change-point models", **24th Signal Processing and Communication Application Conference (SIU)**, Zonguldak, Türkiye, 1301–1304, 16-19 Mayıs 2016.

[11] D. Erhan, E. Anarım, ve G. K. Kurt, "DDoS attack detection using matching pursuit algorithm", **24th Signal Processing and Communication Application Conference (SIU)**, Zonguldak, Türkiye, 1081–1084, 16-19 Mayıs 2016.

[12] S. Shakya ve B. R. Kaphle, "Intrusion Detection System Using Back Propagation Algorithm and Compare its Performance with Self Organizing Map", *J. Adv. Coll. Eng. Manag.*, 1(0), 127–138, 2016.

[13] Ç. Kaya, O. Yıldız, ve S. Ay, "Performance analysis of machine learning techniques in intrusion detection", **24th Signal Processing and Communication Application Conference (SIU)**, Zonguldak, Türkiye, 1473–1476, 16-19 Mayıs 2016.

[14] Ç. Kaya ve O. Yildiz, "Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz", *Marmara Fen Bilim. Derg.*, 26(3), 89–104, 2014.

[15] S. Mukkamala, A. H. Sung, ve A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms", *J. Netw. Comput. Appl.*, 28(2), 167–182, 2005.

[16] S. Peddabachigari, A. Abraham, C. Grosan, ve J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", *J. Netw. Comput. Appl.*, 30(1), 114–132, 2007.

[17] S. S. Sivatha Sindhu, S. Geetha, ve A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach", *Expert Syst. Appl.*, 39(1), 129–141, 2012.

[18] S.-J. Horng vd., "A novel intrusion detection system based on hierarchical clustering and support vector machines", *Expert Syst. Appl.*, 38(1), 306–313, 2011.

[19] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, ve N. Yazdani, "Mutual information-based feature selection for intrusion detection systems", *J. Netw. Comput. Appl.*, 34(4), 1184–1199, 2011.

[20] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, ve K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method", *Expert Syst. Appl.*, 39(1), 424–430, 2012.

[21] M. A. Ambusaidi, X. He, P. Nanda, ve Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm", *IEEE Trans. Comput.*, 65(10), 2986–2998, 2016.

[22] Y. Zhu, J. Liang, J. Chen, ve Z. Ming, "An improved NSGA-III algorithm for feature selection used in intrusion detection", *Knowl.-Based Syst.*, 116, 74–85, 2017.

[23] S. Singh, S. Silakari, ve R. Patel, "An efficient feature reduction technique for intrusion detection system", **International Conference on Machine Learning and Computing**, Baoding, Çin, 147-153, 12-15 Temmuz 2009.

[24] P. R. K. Varma, V. V. Kumari, ve S. S. Kumar, "Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System", *Procedia Comput. Sci.*, 85, 503–510, 2016.

[25] Haltaş, A, Alkan, A. "Medlıne Veritabanı Üzerinde Bulunan Tıbbi Dökümanların Kanser Türlerine Göre Otomatik Sınıflandırılması". *Bilişim Teknolojileri Dergisi*, 9(2), 181-186, 2016.

[26] G.-B. Huang, Q.-Y. Zhu, ve C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, 70(1–3), 489–501, 2006.

[27] T. Kavzoğlu, E. K. Şahin, ve İ. Çölkesen, "Heyelan Duyarlılık Analizinde Ki-Kare Testine Dayalı Faktör Seçimi", **V. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu (UZAL CBS 2014)**, İstanbul, Türkiye, 14-17 Ekim 2014.

[28] K. Kira ve L. A. Rendell, "A pratical approach to feature selection", **Ninth international workshop on Machine learning**, Scotland, İngiltere, 249-256, 1-3 Temmuz 1992.

[29] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF", **Machine Learning: ECML-94**, Catania, İtalya, 171–182, 6-8 Nisan 1994.

[30] Internet: KDD Cup 99 dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 10.03.2017.

[31] Internet: Precision and Recall. https://en.wikipedia.org/wiki/Precision_and_recall, 21.09.2017.