Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması

Atilla Özgür[1], Hamit Erdem[2]

[1]İŞKUR, Ankara, Türkiye
[2]Elektrik Mühendisliği Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye
ati.ozgur@gmail.com, herdem@baskent.edu.tr
(Geliş/Received: 31.05.2012; Kabul/Accepted: 12.09.2012)

Özet— Bilgisayar sistemlerinin yaygınlaşmasıyla, sistemlerin güvenliğini sağlayan araçlar önem kazanmaktadır. Bu araçların en önemlilerinden biri olan Saldırı Tespit Sistemlerine (STS) duyulan önem artmaktadır. Bu sistemlerin eğitiminde geleneksel yöntemlerin yanı sıra, çok sayıda Makine Öğrenme yöntemi kullanılabilir. Makine Öğrenme yöntemlerinin bu alanda etkinliğini araştırmak için, bu çalışmada en çok kullanılan 4 yöntemin başarısı test edilmiştir.
Bu çalışmada akademik STS araştırmalarında en çok kullanılan KDD99 veri seti kullanılmıştır. Veri seti üzerinde yaygın kullanılan dört algoritmanın performansları, Weka Makine Öğrenmesi Tezgahı kullanılarak karşılaştırılmıştır. Kullanılan algoritmalar: Karar Ağaçları, Yapay Sinir Ağları, Destek Vektör Makineleri ve AdaBoost yöntemleridir.

Anahtar Kelimeler— Güvenlik, saldırı tespit sistemleri, makine öğrenmesi, kdd99 karar ağaçları, destek vektör makineleri, yapay sinir ağları, adaboost

Comparison of Out-of-Box Machine Learning Algorithms used in Intrusion Detection Systems

Abstract— Security tools becomes more and more important as computers systems becomes common in Enterprises and normal life and security of these systems becomes an important issue. Need for Intrusion Detection Systems increases since they are among the most important security tools in use. A lot of different Machine Learning algorithms can be used in IDS. This work uses KDD99 as its dataset since it is the most used data set in IDS research. Four common algorithms are compared on this data set using Weka , Machine Learning Toolbox. Compared Algorithms are: Decision Trees, Neural Networks, Support Vector Machines and AdaBoost. According to results ; we conclude that using training KDD99 dataset without any preprocessing gives too good results. Even though Intrusion Detection System's performance on KDD99 dataset should be taken into account, this performance is not gold standard, and should not be thought as Intrusion Detection System will perform equally well in real system.

Keywords— Security, intrusion detection systems, machine learning, kdd99, decision trees, support vector machines, neural network, adaboost

1. GİRİŞ

Gelişen global ekonomi dünyasında Bilgi Teknolojileri (BT, İng: Information Technology - IT) kurumlar ve devletler için merkezi bir öneme sahip olmuştur. BT kullanımının artması ile birlikte BT güvenliği de gittikçe önem kazanmıştır. Saldırı Tespit Sistemleri (STS) her BT güvenlik sisteminin bir parçasıdır ve BT güvenliğinde önemli bir yere sahiptir. CSI Anketine [1] göre, BT içinde STS kullanımı %60'dan fazladır. Bu duruma rağmen STS, araştırmacılar için hala çözülmemiş bir problemdir [2].

STS sistemleri yapılan saldırıları normal durumlardan ayırt etmelerine göre ikiye ayrılırlar. İmza tabanlı ve Anormallik tabanlı [3]. İmza tabanlı STSler , virüs tanıma sistemlerine benzerler. Bilinen saldırıları veri tabanlarında saklayarak gelen örnekleri bu veri tabanına bakarak sınıflandırırlar. Anormallik tabanlı STSler ise ilk önce neyin normal olduğu tanımını yaparak başlarlar, arkasından gelen örnekleri bu tanıma göre normal veya anormal (saldırı) olarak sınıflandırırlar.

İmza tabanlı sistemler ile anormallik tabanlı sistemler arasındaki en önemli fark, ilk defa ortaya çıkan veya sistemin daha önce görmediği saldırılara karşı davranışlarıdır. İmza tabanlı yöntemler sadece bilinen saldırıları bulabilirken, anormallik tabanlı yöntemler ise en yeni, daha önce görülmemiş saldırıları bulabilirler. Bu özelliklerine rağmen, anormallik tabanlı sistemlerin yanlış alarm oranı, imza tabanlı sistemlere göre çok yüksek olduğundan, endüstride daha çok imza tabanlı yöntemler kullanılmaktadır.

Bunun yanında akademi tarafında, yeni saldırıları bulma özelliklerinden dolayı anormallik tabanlı yöntemler tercih edilmektedir [2,4]. STS ve Anormallik tabanlı sistemler konusunda Tavallae ve diğerleri [4] 2000-2008 yılları arasından çoğunluğu indeks edilen dergilerden olan 276 çalışma incelemiştir. Bu çalışmalarda en çok kullanılan veri seti KDD99 veri setidir [5].

Bu araştırmanın amacı; hiç bir önişleme yapılmamış KDD99 veri setinin STS araştırmalarında tek başına kullanılmasının, önerilen STS'nin gerçek sistemlerde kullanılabilecek durumda olmadığını göstermektir. Bu araştırmanın daha önce tek bir makine öğrenme yöntemini, J4.8 Karar Ağaçlarını, kullanan sürümü [6] yayınlanmıştır.

Anormallik tabanlı sistemlerde bir çok makine öğrenmesi ve veri madenciliği yöntemi kullanılmaktadır. Daha önceki çalışmada karar ağaçları en çok kullanılan algoritmalar arasında olduğu için seçilmişti [6].

Bu çalışmada yine en çok kullanılan algoritmalar [7] arasından Karar Ağaçları, Destek Vektör Makineleri, Yapay Sinir Ağları ve Adaboost seçilmiştir. Bu algoritmaların çalışma ortamı için Weka Makine Öğrenmesi Tezgahı kullanılmıştır [8] .

Makine Öğrenme Algoritmalarının karşılaştırması sadece kesinlik (Accuracy, Detection Rate) kullanılarak yapılmıştır. Makine Öğrenme algoritmalarının karşılaştırılmasında daha iyi yöntemler mevcuttur [9]. Ama bu çalışmada bilerek sadece kesinlik kullanılmıştır. Karşılaştırma kıstası olarak, sadece kesinlik kullanılması, kullanılan algoritmaların olduklarından daha iyi görünmelerine neden olmaktadır.

Makalede sırasıyla aşağıdakiler anlatılacaktır. İkinci bölümde saldırı tespit ihtiyacı ve Saldırı Tespit Sistemlerinin tanımı verilecektir. Üçüncü bölümde kullanılan makine öğrenme algoritmaları anlatılacaktır.

Dördüncü bölümde KDD99 veri setinin özellikleri anlatılacaktır. Bu konuyla ilgili diğer araştırmalar beşinci bölümde verilecektir. Çok kullanılan makine öğrenme algoritmaları, KDD99 %10 eğitim veri seti üstünde eğitildikten sonra; KDD99 %10 eğitim, KDD99 tam eğitim ve KDD99 test veri setleri üzerinde denenecektir. Tartışma bölümünde, elde edilen sonuçlara göre karşılaştırma tartışılacaktır.

2. SALDIRI TESPİT SİSTEMLERİ

Bilgisayar Virüsleri ile ilgili teorik altyapı Neumann tarafından "Theory of self-reproducing automata" makalesi ile 1966'da ilk defa verilmiştir [10]. 1971'deki bilinen ilk virüsten 2010 yılına kadar, güvenlik açıkları boyut ve maliyet olarak artmıştır. Devlet kurumlarına ve şirketlere bu güvenlik açıklarının maliyeti milyon dolarlarla ölçülmektedir. Hobi olarak başlayan bilgisayar kırma işlemleri (Cracking), holdingler, mafya ve ordular için bir araca dönüşmeye başlamıştır [11-16]. Bu saldırılar Siber Saldırılar olarak tanımlanır.

Siber Saldırıların zamana göre değişimi şekil 1'de görülebilir. Şekil 1'deki saldırı değişimine göre, siber saldırıların gelecekte azalmayacağı görülmektedir. Bilişim suçlarındaki benzer bir artış Dicle ve Doğan tarafından rapor edilmiştir [17]. Bundan dolayı güvenlik ürünleri ve özellikle Saldırı Tespit Sistemleri daha da önem kazanacaktır.

2.1. Saldırı Tespit Sistemleri Tanımı

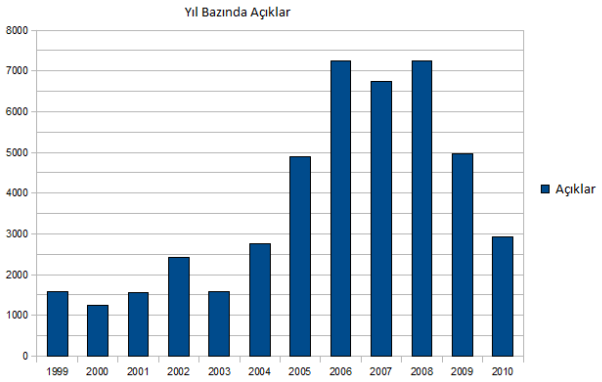
Saldırı Tespit ve Önleme Sistemleri Rehberi (Guide to Intrusion Detection and Prevention Systems) raporuna göre STS tanımı aşağıdaki gibidir:"Saldırı Tespit bilgisayar veya ağ sistemi üzerindeki olayları gözleme ve bu olayları analiz ederek olası vakaları bulma işlemidir. Bu vakalar, bilgisayar güvenlik politikalarının, kabul edilebilir kullanım politikalarının veya standart güvenlik pratiklerinin ihlali veya ihlal edilmesi tehdididir [3]."

Saldırı Tespit Sistemleri çoğu zaman iki yöntemle sınıflandırılırlar.

- Sistemlerin Kurulumuna Göre
  - Ağ STS (Network IDS)
  - Ev-Sahibi STS (Host IDS)
- Sistemin Tespit Mantığına Göre
  - İmza tabanlı STS
  - Anormallik STS, (Signature based IDS and Anomaly based IDS)

İmza tabanlı STSler, virüs tanıma sistemlerine benzerler. Bilinen saldırıları veri tabanlarında saklayarak gelen örnekleri bu veri tabanına bakarak sınıflandırırlar. Anormallik tabanlı STSler ise ilk önce neyin normal olduğu tanımını yaparak başlarlar. Arkasından gelen örnekleri bu tanıma göre normal veya anormal (saldırı) olarak sınıflandırırlar.

Saldırı Tespit klasik sınıflandırma problemi olarak düşünülebilir. Örneğin verilen bir ağ bağlantısının özellikleri kullanılarak, verilen ağ bağlantısı örneğini normal veya saldırı olarak sınıflandırabiliriz.



Şekil 1. CVE yıllara göre yaygın olarak bulunan açıklar

3. MAKİNE ÖĞRENME SINIFLANDIRMA ALGORİTMALARI

Bu çalışmada daha öncede belirtildiği gibi en çok kullanılan algoritmalar [7] arasında bulunan Karar Ağaçları, Destek Vektör Makineleri, Yapay Sinir Ağları ve AdaBoost kullanılmıştır.

3.1. Karar Ağaçları - Sınıflandırma Ağaçları (Decision Tree - Classification Tree)

Karar Ağaçları veya Sınıflandırma Ağaçları, İstatistiksel öğrenme ve Veri Madenciliğinde çok bilinen sınıflandırma metodudur. Veri Madenciliğinde çok kullanıldığından dolayı [18], Veri Madenciliğinin yük beygiri [19] olarak ta adlandırılmıştır.

Bunun nedenleri arasında:

- Karar Ağaçlarının eğitimi ve test edilmesi çok hızlıdır.
- Karar Ağaçlarının sonuçlarının insanlar tarafından yorumlanması çok kolaydır.
- Karar Ağaçlarının sonuçları daha rahat görselleştirilebilir.
- Karar Ağaçlarının sonuçları kural çıkarımında kullanılabilir.

3.2. Yapay Sinir Ağları (Artificial Neural Networks)

Yapay Sinir Ağları (YSA), temellerini biyolojik sinir hücrelerinden (nöron) alan, güçlü bir sınıflandırma aracıdır. En temel yapı taşı olarak nöronları kullanırlar. Yapay nöronlar biyolojik nöronlara benzer olması için tasarlanmıştır. Yapay bir nöronun 3 özelliği vardır.

1. girişler
2. toplama birimi
3. transfer fonksiyonu

Nörona verilen girişler, ağırlıklar ile çarpıldıktan sonra toplama biriminde toplanırlar. Bulunan toplam, transfer fonksiyonuna giriş olarak verilir. Transfer fonksiyonun çıkış değeri, o nöronun çıkış değeri olarak alınır. Yapay Sinir Ağları mimarisinde bir çok transfer fonksiyonu kullanılmaktadır. Örneğin sigmoid fonksiyonu sınıflandırma için kullanılırken, gaussian (normal) fonksiyonu, fonksiyon yakınlaştırma için kullanılabilir.

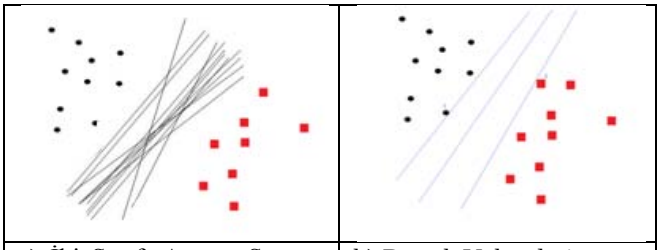
Normalde tek bir nöron sadece doğrusal problemleri çözebilir. Diğer bir çok sınıflandırma problemini çözebilmek için çok katmanlı yapay sinir ağları, (Multi Layer Perceptron-MLP) kullanılmaktadır. Çok katmanlı YSA, fonksiyon uydurma, sınıflandırma ve eşleşme problemlerinde sıkça kullanılmaktadır. Sınıflandırmadaki başarısından dolayı STS'lerde kullanılmıştır [20]. STS'ler haricinde kullanımına bir örnek için [21].

3.3. Toplama Yöntemi (Ensemble Learning, Boosting-AdaBoost)

Boosting algoritmaları toplama yöntemi (ensemble method) denilen Makine Öğrenme Algoritmalarıdır. Toplama yöntemi algoritmaları bir çok güçsüz öğrenme algoritmasının birleştirilerek daha güçlü bir öğrenme algoritması oluşturulmasıdır. Güçsüz algoritma olarak genellikle kolay oluşturuldukları için karar ağaçlarının hızlı türetilen versiyonları seçilir. Boosting diğer güçsüz sınıflandırıcılar ile çalışan toplama yöntemi algoritmalarına benzemektedir. Yine aynı şekilde güçlü bir sınıflandırıcı, çok sayıda güçsüz sınıflandırıcının birleştirilmesi ile oluşturulmaktadır. Ama bu oluşturulma işlemi yapılırken modelin daha önce yanlış sınıflandırdığı örnekleri daha iyi sınıflandırması için her güçsüz sınıflandırıcıya bir ağırlık atanmaktadır. Bundan dolayı öğrenilen modellerin ağırlıkları, bir örnekten diğerine değişiklik göstermektedir. AdaBoost en çok kullanılan boosting algoritmaları arasındadır [22,23].

3.4. Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makineleri yüksek performans gösteren ve oldukça basit bir fikir ile oluşturulmuş bir öğrenme algoritmasıdır [24]. Temelini İstatiksel Öğrenme teorilerinden alan, Destek Vektör Makineleri teori olarak iki sınıfa ait örnekleri doğrusal olarak en iyi ayıran destek vektörlerine dayanmaktadır. Şekil 2a üzerinde görülebileceği gibi, iki sınıfı ayıran sonsuz sayıda doğru bulunabilir. Ama bunları en iyi ayıran doğru 2b, 2 sınıfa da belli bir uzaklıkta olan doğrudur. Bu fikir kullanarak oluşturulan Destek Vektör Makineleri son derece başarılı sınıflandırıcılardır. Destek Vektör Makineleri birçok sahada kullanılmaktadırlar. STS sahasında sınıflandırıcı olarak kullanılmıştır [25].



a) İki Sınıfı Ayıran Sonsuz Sayıda Doğru | b) Destek Vektörleri

Şekil 2. Destek vektör makineleri teorisi

4. VERİ SETİ

KDD-DARPA veri seti Amerikan Hava Kuvvetleri (US Air Force) network ağına benzer bir yapıya sahip olması düşünülerek tasarlanmış, bir simülasyon veri setidir. DARPA tarafından sponsor edilen ilk çalışma MIT Lincoln Lab'ı tarafından 1998 yılında tamamlanmıştır [26].

Bir yıl sonra aynı çalışma Bilgisayar Güvenliği ile çalışanlardan alınan yorumlar ile iyileştirmeler yapılarak tekrarlanmıştır. Aynı veri seti KDD tarafından her sene

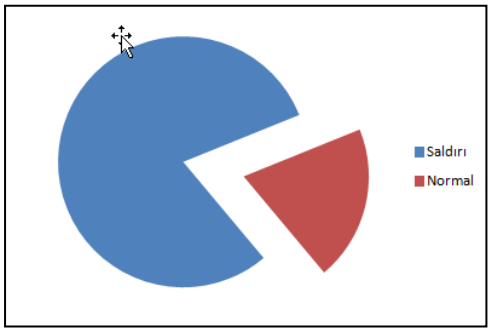
tekrarlanan yarışma için seçilmiştir. KDD yarışmasından sonra, Saldırı Tespit Sistemleri üzerine olan araştırmalar, özellikle Veri Madenciliği ve Makine Öğrenme konusunda çok artmıştır. Her ne kadar KDD99'un problemleri çok iyi biliniyor olsa da [27,28] KDD99-DARPA veri seti Saldırı Tespit Veri Madenciliği çalışmalarında en çok kullanılan veri setidir [4].

Tavallaee ve diğerleri [4], çoğunluğu indeks edilen dergilerden alınmak üzere 276 makale incelemiştir. Tavallaee ve diğerlerinin tarama çalışmasına göre inceledikleri 276 çalışmadan 209'u (%75) KDD99-DARPA kullanmıştır. Yine çok kullanılan UNM veri seti [16]'de sayılırsa, sadece %6 kendi veri setini kullanmıştır. İncelenen makalelerin dağılımı tablo 1'deki gibidir.
Bir çalışmada birden fazla veri seti kullanılabileceğinden toplam sayı 276'dan fazla çıkmaktadır.

Tablo 1. Tavallaee ve diğerlerinin tarama çalışmasına göre veri seti kullanımları

Veri Seti Adı	Kullanım Sayısı	Kullanım Yüzdesi
DARPA	67	%24
KDD99	77	%28
KDD99-DARPA + Ek Saldırı	41	%15
Veri Seti Belirtilmemiş	86	%31
Başka Veri Seti	16	%6

DK99C (DARPA-KDD99 Cup Veri Seti) önceden işlenmiş bir veri setidir, detaylar için bakınız [29]. KDD99 Eğitim veri seti 4.8 milyon civarı kayıttan oluşmaktadır. Bu kayıtlar 22 saldırı ve normal örneklerden oluşmaktadır. Bu veri seti çok dengesiz bir veri setidir, kayıtların yaklaşık %80'ı saldırıdır (Şekil 3).



Şekil 3. KDD99 saldırı normal oranı

Bu veri setindeki her saldırı 4 majör gruba dahildir.

- R2L - Root to Local
- U2R - User to Root
- DOS - Denial of Service
- Probe - Probing attacks

Saldırıların neptune, smurf, perl gibi isimleri vardır her biri yukarıdaki 4 gruptan birine dahildir. Bu veri setinde kullanılan saldırılar için daha fazla bilgi almak istiyorsanız Kendal'ın [30] yüksek lisans tezine bakınız.

Her ne kadar weka [19] birçok değişik format ile çalışabilse de, en iyi şekilde kendi formatı olan arff ile çalışmaktadır.KDD99'un verildiği format olan C4.5 formatından arff formatına daha önceki bir çalışmada geçiş yapılmıştır [31].

5. İLGİLİ ÇALIŞMALAR

Wu ve diğerleri [32] Karar Ağaçlarını çalışmalarında kullanmışlardır. C4.5 ile Destek Vektör Makinelerini karşılaştırmışlardır. KDD99'ın %10 eğitim versiyonunu kullanmışlardır. Yaptıkları ön işleme aşağıdaki gibidir. Normal dağılıma bakarak, verinin her 10.000'lik grubunu, 10%, 20%, 30%, . . ., 90% olacak şekilde seçmiş, geri kalan veriyi, yani saldırı verisini normalleştirmiş ve örneklemişlerdir.

Sonuçlarını SVM,KDD99 Yarışması Birincisinin Metodu ve Karar Ağaçları arasında karşılaştırmışlardır. Karşılaştırma için Kesinlik ve Sahte Alarm sonuçları gösterilmiştir. Buldukları sonuçlara göre Karar Ağaçlarının bir çok durumda en iyi olduğuna karar vermişlerdir. 10 farklı deney yaptıklarından dolayı %40 ile %90 arasında çok farklı sonuçlar elde etmişlerdir.

Quan ve diğerlerinin çalışması [33] çoğunlukla KDD-DARPA veri setinin dengesiz olması ile ilgilidir. Karar Ağaçları (C4.5 algoritması) fazla örnekleme (oversampling) yapılmasından çok az etkilenmektedir ama az örnekleme (undersampling) yapılmasından daha fazla etkilenmektedir. Bu probleme çözüm bulmak için ve daha az olan saldırıları tespit etmek için, C4.5'e bir uyarlama önermektedirler. Bu uyarlamayı PC4.5 (Parametrized) olarak adlandırmışlardır.

KDD99 %10 veri setini kullanarak deneylerini yapmışlardır. İki katmanlı bir mimarileri vardır. İlk katman saldırı sınıfları ve normal arasında en çok örneği olanları (Normal, smurf,neptune) tespit etmeye çalışmaktadır. Diğer saldırılarda karışık olarak tespit edilmeye çalışılmaktadır. Bu katmanın çalışmasında, C4.5'ın diğer makine öğrenme algoritmalarından daha iyi olduğunu bulmuşlardır. İlk katmanda, C4.5, Naive Bayes, BayesNet, SVM, KNN algoritmaları kullanılmıştır. 10'lu çapraz-geçerlilik ile değerlendirmeler yapılmıştır. Sonuçlara göre C4.5, bu veri seti üzerinde diğerlerine göre daha iyi bir algoritmadır.

İkinci katmanda, karışık saldırı olarak birleştirilenlerden 20 saldırı sınıfını bulmaya çalışmaktadırlar. Kendi önerdikleri PC4.5 algoritması ile C4.5 algoritmasının karşılaştırmasını yapmışlardır. Sonuçlarında kesinlik, f-ölçümü, karar ağaçlarındaki yaprak sayısını vermişlerdir. Toplam kesinlik sonucunu vermemişlerdir ama her saldırı için ayrı ayrı kesinlik verilmiştir. Bu kesinlik değer %58'den %100'e kadar değişmektedir. DOS ve Probe saldırılarında, %95 üzerinde çok iyi sonuçlar raporlamışlardır.

Sabhani ve Serpen [34] neden makine öğrenme algoritmalarının, DOS ve PROBE saldırıları dışında, KDD99 veri setinde başarısız olduğunu göstermişlerdir.

DOS ve Probe saldırıları KDD99 Test ve Eğitim veri setlerinde istatistiksel olarak birbirine benzemektedir.
Ama diğer saldırılar, test ve eğitim setlerinde bir birlerinden çok farklıdır. Literatür taraması yaparak Makine Öğrenme Algoritmalarının genel olarak bu veri setinde başarılı olduğunu ama R2L ve U2R saldırılarında başarısız olduğunu göstermişlerdir.

Lee ve diğerleri [35] yaptığı çalışmada, DARPA tcp dump verisini işlemiştir. KDD99 aynı verinin ön işlemesi ile elde edildiğinden dolayı, teorik olarak aynı özellikleri kullanmaları gerekir. Özellik olarak sadece tek bir bağlantıdan elde edilebilenleri kullanmış, çoklu bağlantı sonucu elde edilebilen özelliklerle ilgilenmemişlerdir.
Bu tür bir ön işleme DARPA içindeki tüm saldırıları yakalamak için yeterli değildir [29].

Saldırı Tespit konusunda özellik çıkarma konusu için bakınız [36]. Kendi sonuçlarını, kesinlik oranı, Lee ve Stolfo [29] ile karşılaştırmışlardır. Çalışmaları DOS ve Probe saldırılarında ilerleme göstermesine rağmen R2L ve U2R saldırılarında kaybetmektedir. Lee ve Stolfo'nun [29] çalışmasında anlatıldığı gibi çoklu bağlantı özellikleri olmadan R2L ve U2R saldırılarını tespit etmek çok zordur.

Folino et al [37] Genetik programlamayı toplama yöntemi için kullanmışlardır. Dağıtık Yapıda bir STS önermişlerdir. Bu dağıtık yapıdaki her düğüm noktası , bir ada noktası olarak sınıflandırıcılardan oluşmaktadır. Her ada düğüm noktasındaki genetik program karar ağacı sınıflandırıcıları üretmektedir. Her sınıflandırıcı düğümdeki kendi verisi ile çalışmaktadır. En iyi genler diğer sınıflandırıcılar ile paylaşılmaktadır. Bütün sınıflandırıcılar oluşturulduktan sonra bir Genetik Program Toplama Yöntemi, Adaboost algoritmasını kullanarak sınıflandırıcılarını birleştirilmektedirler.

KDD99 %10 verisi ile çalışmışlardır. Kendi algoritmalarını KDD99 yarışmasını birinci ve ikinci bitiren algoritmalar ile karşılaştırmışlardır. Algoritmaları Kesinlik, Yanlış Alarm ve ROC eğrileri olarak daha iyi sonuçlar göstermiştir. Averaj sonuçları KDD99 birinci ve ikincisine yaklaşsa da onlardan daha iyi değildir.

6. DENEYSEL SONUÇLAR

Sınıflandırıcı Algoritmaları %10 veri setinde eğiterek model oluşturuyoruz. Daha sonra bu modeli kullanarak 3 veri setinde test yapıyoruz. Elde ettiğimiz sonuçların özet bilgileri tablo 2-5'te görülebilir. Karar Ağaçları ile yapılan deney sonuçları daha önce başka bir çalışmada [6] sunulmuştur.

7. TARTIŞMA

Makine Öğrenme algoritmaları KDD99 Eğitim %10 veri setinde eğitildikten sonra, elde edilen modeller KDD99 Eğitim %10, KDD99 Tüm Eğitim seti ve KDD99 Test seti üzerinde denenmiştir. Bu deneylerden elde edilen özet sonuçlar tablo 6'da görülmektedir.
İlk kolondaki %10 Eğitim sonuçları çok iyi çıkmaktadır.

Bu sonuçlar sınıflandırıcılarımızın eğitim setini nerdeyse ezberlediklerini gösteriyor. Ama sınıflandırıcılarımız her ne kadar ezberleme yapsa da, test veri seti üstünde çok iyi başarı (%99+ ve %91+) gösteriyorlar. Kullanacağımız amaca göre bu dört sınıflandırma algoritması da çok iyi sonuçlar vermektedir. Hatta algoritmalarda nerdeyse hiç bir ayar yapılmadığı düşünülürse çok çok iyi sonuçlar alınmaktadır.

Sınıflandırıcılarımız %10 eğitim setinde eğitilmelerine rağmen %100 olan Eğitim Veri Seti Tüm ve Test Veri Seti üzerinde de çok iyi başarı göstermektedirler. Bunun nedeni 3 veri setinde de simülasyon saldırıların kullanılmış olması ve bazı saldırılar dışında, çoğunluk olarak kullanılan saldırıların birbirlerine çok benziyor olmasıdır [30]. Bu durum KDD99-DARPA veri setinin simülasyon özelliğini çok iyi bir şekilde ortaya koymaktadır.

Tablo 6. Makine öğrenmesi - 4 algoritmanın kesinlik sonuçları özeti

	Eğitim Veri Seti %10	Eğitim Veri Seti Tümü	Test Veri Seti
Karar Ağaçları	99.9818%	99.9823 %	91.7406 %
Yapay Sinir Ağları	99.9328%	99.9517 %	92.4821 %
Destek Vektör Makineleri	99.8516%	99.8741 %	92.3049 %
AdaBoost	98.2792%	99.0101 %	91.3886 %

8. SONUÇLAR

KDD99 veri seti üstünde %90 üstünde Kesinlik elde etmenin çok kolay olduğunu gösterdik. Kolay erişilebilen ve üzerinde hiç bir ayar yapmadan çalıştırılan bu makine öğrenme algoritmaları KDD99'un simülasyon doğasını iyi göstermektedir. Bu simülasyon doğasından ötürü, KDD99 veri setinin herhangi bir ön işleme olmadan kullanılması çok iyi sonuçlar vermektedir. Kullanılan makine öğrenme yöntemleri arasında belirgin bir fark yoktur. Simülasyon yapısından dolayı, KDD99 Saldırı Tespit Araştırmalarında tek başına kullanılmak için uygun bir veri seti değildir. Bu veri setinde elde edilmiş iyi sonuçlar, gerçek bir ortamda da benzer sonuçlar elde edileceği anlamına gelmemektedir. Her ne kadar Saldırı Tespit Sistemlerinin KDD99 üzerindeki sonuçlarının da göz önünde bulundurulması gerekse de, KDD99 üzerindeki performansları altın standart değildir. Oluşturulan Saldırı Tespit Sisteminin gerçek bir ortamda benzer bir sonuç vereceği düşünülmemelidir.

KDD99 veri seti üstünde elde edilen sonuçlar başka yöntemler, diğer veri setlerindeki sonuçlar ile desteklenmelidir.

Tablo 2. J48 sınıflandırıcının özet sonuçları

	Eğitim Veri Seti %10	Eğitim Veri Seti Tümü	Test Veri Seti
Toplam Örnek Sayısı	494021	4898431	311029
Doğru Sınıflandırılan Örnek Sayısı	493931	4897566	285340
Doğru Sınıflandırılan Örnek Yüzdesi	99.9818%	99.9823%	91.7406%
Yanlış Sınıflandırılan Örnek Sayısı	90	865	25689
Yanlış Sınıflandırılan Örnek Yüzdesi	0.0182%	0.0177%	8.2594%
Kappa İstatistik	0.9997	0.9997	0.87
Ortalama Hata	0	0	0.0041
Kök Ortalama Kare Hatası	0.0029	0.0028	0.0641
Olayların Kapsanması (0.95 level)	99.9874%	99.986%	91.7442%

Tablo 3. MLP yapay sinir ağları sınıflandırıcının özet sonuçları

	Eğitim Veri Seti %10	**Eğitim Veri Seti Tümü**	**Test Veri Seti**
Toplam Örnek Sayısı	494021	4898431	311029
Doğru Sınıflandırılan Örnek Sayısı	493689	4896065	287646
Doğru Sınıflandırılan Örnek Yüzdesi	99.9328%	99.9517 %	92.4821 %
Yanlış Sınıflandırılan Örnek Sayısı	332	2366	23383
Yanlış Sınıflandırılan Örnek Yüzdesi	0.0672 %	0.0483 %	7.5179 %
Kappa İstatistik	0.9979	0.9985	0.7887
Ortalama Hata	0.001	0.0008	0.0762
Kök Ortalama Kare Hatası	0.0258	0.0222	0.2738
Olayların Kapsanması (0.95 level)	99.9453 %	99.9577 %	92.5743 %

Tablo 4. SVM destek vektör makineleri sınıflandırıcının özet sonuçları

	Eğitim Veri Seti %10	**Eğitim Veri Seti Tümü**	**Test Veri Seti**
Toplam Örnek Sayısı	494021	4898431	311029
Doğru Sınıflandırılan Örnek Sayısı	493288	4892262	287095
Doğru Sınıflandırılan Örnek Yüzdesi	99.8516%	99.8741 %	92.3049 %
Yanlış Sınıflandırılan Örnek Sayısı	733	6169	23934
Yanlış Sınıflandırılan Örnek Yüzdesi	0.1484 %	0.1259 %	7.6951%
Kappa İstatistik	0.9953	0.996	0.7843
Ortalama Hata	0.0015	0.0013	0.077
Kök Ortalama Kare Hatası	0.0385	0.0355	0.2774
Olayların Kapsanması (0.95 level)	99.8516 %	99.8741 %	92.3049%

Tablo 5. AdaBoost sınıflandırıcının özet sonuçları

	Eğitim Veri Seti %10	**Eğitim Veri Seti Tümü**	**Test Veri Seti**
Toplam Örnek Sayısı	494021	4898431	311029
Doğru Sınıflandırılan Örnek Sayısı	485520	4849943	284245
Doğru Sınıflandırılan Örnek Yüzdesi	98.2792%	99.0101 %	91.3886 %
Yanlış Sınıflandırılan Örnek Sayısı	8501	48488	26784
Yanlış Sınıflandırılan Örnek Yüzdesi	1.7208 %	0.9899 %	8.6114%
Kappa İstatistik	0.9463	0.9689	0.7601
Ortalama Hata	0.0212	0.0159	0.0877
Kök Ortalama Kare Hatası	0.11	0.0907	0.2759
Olayların Kapsanması (0.95 level)	99.9512 %	99.9659 %	93.879%

KAYNAKLAR

[1] CSI, 2010-2011 Computer Crime and Security Survey, *CSI*, **Tech. report,** 2011.

[2] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection", Washington, DC, USA, pp. 305-316, 2010. [Online]. http://dx.doi.org/10.1109/SP.2010.25

[3] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (idps)", *NIST Special Publication*, vol. 800, no. 2007, p. 94, 2007.

[4] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods", Systems, Man, and Cybernetics, Part C: *Applications and Reviews, IEEE Transactions on*, vol. 40, no. 5, pp. 516-524, sept 2010.

[5] K. C., 1999, KDD Cup 1999 Data Task Description.

[6] A. Özgür and H. Erdem, "An Application of Decision Trees in Intrusion Detection", V. **International Information Security and Cryptology Conference - ISCTURKEY 2012**, Ankara, Turkey, May 17-18, 2012.

[7] X. Wu et al., "Top 10 algorithms in data mining", *Knowledge and Information Systems*, vol. 14, pp. 1-37, 2008.

[8] R. R. Bouckaert et al., "WEKA--experiences with a java opensource project", *Journal of Machine Learning Research*, vol. 11, pp. 2533-2541, 2010.

[9] T. Fawcett, **ROC graphs: Notes and practical considerations for researchers**, HP Labs, Tech. rep. 2004.

[10] J. V. Neumann, **Theory of Self-Reproducing Automata**, A. W. Burks, Ed. Champaign, IL, USA: University of Illinois Press, 1966.

[11] T. M. Chen and J. M. Robert, "The Evolution of Viruses and worms", *Statistical methods in computer security*, 2004.

[12] InfoWorldSecurityCentralNews, Military contractors now targeted by Chinese cyber attacks, F-Secure, 2010.

[13] R. McMillan, With botnets everywhere, DDoS attacks get cheaper, 2009.

[14] MsnbcNews, Lockheed Martin says it thwarted 'tenacious' cyber attack, June 2011.

[15] İnternet: Wikipedia, Timeline of computer viruses and worms, http://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms, 2010.

[16] N. Shachtman, Exclusive Computer Virus Hits U.S. Drone Fleet, 2011.

[17] H. Dijle and N. Doğan, "Türkiye'de Bilişim suçlarına Eğitimli İnsanların Bakışı", *Bilişim Teknolojileri Dergisi*, vol. 4-2, 2011.

[18] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review", *Applied Soft Computing*, vol. 10, no. 1, pp. 1-35, 2010. [Online]. http://www.sciencedirect.com/science/article/B6W86-4WV15J9-1/2/07117ede6d9ef58ed75bf405560da6d9

[19] I. H. Witten and E. Frank, **Data Mining: Practical machine learning tools and techniques (Third Edition)**: Morgan Kaufmann Pub, 2011.

[20] C. Bitter, D. A. Elizondo, and T. Watson, "Application of artificial neural networks and related techniques to intrusion detection", pp. 1-8, july 2010.

[21] K. K. Çevik and E. Dandıl, "Yapay Sinir Ağları İçin Net Platformunda Görsel Bir Eğitim Yazılımının Geliştirilmesi", *Bilişim Teknolojileri Dergisi*, vol. 5-1, 2012.

[22] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting", *Journal-Japanese Socıety For Artıfıcıal Intelligence*, vol. 14, pp. 771-780, 1999.

[23] R. E. Schapire, "A brief introduction to boosting", San Francisco, CA, USA, pp. 1401-1406, 1999. [Online]. http://portal.acm.org/citation.cfm?id=1624312.1624417

[24] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines", *Intelligent Systems and their Applications*, IEEE, vol. 13, no. 4, pp. 18-28, 1998.

[25] W.H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection", *Computers&Operations Research*, vol. 32, no. 10, pp. 2617-2634, 2005, Applications of Neural Networks. [Online]. http://www.sciencedirect.com/science/article/pii/S0305054804000711

[26] R. K. Cunningham et al., **Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation**, Massachusetts Inst Of Tech Lexıngton Lıncoln Lab., Tech. rep. 1999.

[27] S. T. Brugger, "KDD Cup'99 dataset (Network Intrusion) considered harmful", *KDnuggets newsletter*, vol. 7, no. 18, p. 15, 2007.

[28] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory", *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262-294, 2000.

[29] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems", *ACM Trans. Inf. Syst. Secur.*, vol. 3, pp. 227-261, November 2000. [Online]. http://doi.acm.org/10.1145/382912.382914

[30] K. Kendall, **A database of computer attacks for the evaluation of intrusion detection systems**, MIT - Massachusetts Institute of Technology, Master's thesis, 1999.

[31] A. Özgür, **Converting KDD99 C4.5 Dataset To Weka Arff**, Baskent University, Tech. rep. 2012.

[32] S. Y. Wu and E. Yen, "Data mining-based intrusion detectors", *Expert Systems with Applications*, vol. 36, no. 3, pp. 5605-5612, 2009.

[33] Z. Quan, G. L. gang, W. C. jun, W. jun, and C. S. fu, "Using an improved C4.5 for imbalanced dataset of intrusion", New York, NY, USA, pp. 67:1--67:4, 2006. [Online]. http://doi.acm.org/10.1145/1501434.1501513

[34] M. Sabhnani and G. Serpen, "Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set", *Intell. Data Anal.*, vol. 8, pp. 403-415, September 2004. [Online]. http://portal.acm.org/citation.cfm?id=1293805.1293811

[35] J.-H. Lee, J.-H. Lee, S.-G. Sohn, J.-H. Ryu, and T.-M. Chung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System", **Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on**, vol. 2, pp. 1170-1175, feb. 2008.

[36] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review", *Computers \& Security*, vol. 30, no. 6-7, pp. 353-375, 2011. [Online]. http://www.sciencedirect.com/science/article/pii/S0167404811000691

[37] G. Folino, C. Pizzuti, and G. Spezzano, "GP Ensemble for Distributed Intrusion Detection Systems", **Pattern Recognition and Data Mining**, S. Singh et al., Eds.: Springer Berlin / Heidelberg, 2005, vol. 3686, pp. 54-62.

[38] X. Bao, T. Xu, and H. Hou, "Network Intrusion Detection Based on Support Vector Machine", **Management and Service Science, 2009. MASS '09. International Conference on**, pp. 1-4, sept. 2009.

[39] G. Folino, C. Pizzuti, and G. Spezzano, "An ensemble-based evolutionary framework for coping with distributed intrusion detection", Genetic Programming and Evolvable Machines, vol. 11, pp. 131-146, 2010, 10.1007/s10710-010-9101-6. [Online]. http://dx.doi.org/10.1007/s10710-010-9101-6

[40] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes", *Proceeding SP '96 Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 120-128, may 1996.