Derin Öğrenme İle Ağ Saldırı Tespiti

Murat Acar, Sheme Hamitaj, Metin Kadık, Ezgi Kaya, Batuhan Meşeci, Rıza Boçoğlu
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
Kocaeli, Turkey
{150201186, 140201113, 150201166, 150201158, 150201180, 150201182}@kocaeli.edu.tr

***Abstract*—Recently, the increase of network attacks has brought forward the need of understanding and detecting these attacks. In this work, we relay on Artificial Neural Networks to make possible the detection of such attacks in real-time. Training was done using the KDD99 data set, an updated version of the KDD98 dataset that contains different intrusions performed in a military network by MIT Lincoln Labs in 1998. This training was done by using selected features of the dataset. The results obtained were compared with other researches in this area.**

***Abstract*—Son yıllarda ağ saldırılarının artması nedeniyle, saldırıların tespiti ve anlamlandırılması ihtiyacı ortaya çıkmıştır. Bu çalışmada Yapay Sinir Ağı kullanılarak KDD99 veri seti ile eğitim yapılmıştır. KDD99 veri seti, 1998 yılında MIT Lincoln Labs tarafından İzinsiz Giriş Tespiti Değerlendirme Programı adı altında Askeri ağ ortamında çeşitli izinsiz girişleri içeren bir veri seti olan KDD98 veri setinin güncellenmiş versiyonudur. Veri setindeki özelliklerden uygun olanları seçilmiş eğitim bu özellikler üzerinden sağlanmıştır. Elde edilen çıktılar bu alanda yapılan diğer çalışmalar ile karşılaştırılmıştır.**

***Index Terms*—Ağ Saldırı Tespiti, YSA, KDD99**

I. GİRİŞ

Bilgisayar ağlarının kullanımı günümüzde devasa boyutlara ulaşmıştır ve gün geçtikçe artmaya devam etmektedir. Ticaret, sağlık ve askeriye gibi kritik alanların yükü bilgisayar ağlarına yüklenmiştir. Bu ağların zaaflarını kullanan kötü niyetli kişilerin sayısı da gün geçtikçe artmaktadır. Bu duruma çözüm olarak Saldırı Tespit Sistemleri (STS) öne çıkmaktadır [9].

STS saldırı esnasında alarm niteliğinde kullanılan yazılım ve donanım bileşenidir. Bir çok bilişim teknoloji sistemlerinin olmazsa olmazlarındandır. Bir saldırı durumu sözkonusu olduğunda sizi uyaracak yegane araçlardan biridir. Saldırı Tespit Sistemlerinde sonuçlar; ataklar ve normal durumlar olarak ikiye ayrılır. Ataklar da tiplerine göre bir çok çeşide ayrılabilir. KDD99 veri setinde 22 adet atak tipi (buffer overflow, neptune, nmap vb.) mevcuttur [8]. Bizim çalışmamızda da sınıflandırma işlemi bu atak tiplerine göre yapılmıştır.

Eğitimde kullanılan model Yapay Sinir Ağı (YSA) dır. İki adet gizli katman ve bir adet çıkış katmanına sahip olan bu yapay sinir ağı KDD99 veri seti üzerinde rahatça kullanılabilmektedir[7].

II. LİTERATÜR İNCELEMESİ

Bu bölümde bu üzerinde yapılmış olan daha önceki çalışmalardan bahsedilecektir.Ayrıca bu konu üzerinde yapılan çalışmalar içinde sadece verikümesi olarak ”KDD99 ”veri kümesi ile yapılmış çalışmaların performans karşılaştırılmaları ele alınacaktır.Böylece bu çalışmada yapılanlar ve literatürde yapılan çalışamalar arasında daha doğru bir karşılaştırma ortamı sağlanmış olur . Sağıroğlu ve arkadaşlarının[10] yaptığı çalışmada önerilen yöntem, literatürde önerilen ve yüksek başarı oranına sahip çalışmalara göre, daha düşük başarı oranına sahip olmasına rağmen literatürde verilen yöntemlerin aksine ağın eğitimi için 22 farklı saldırı kullanmışlar, literatürde önerilen ara katman nöron sayılarının dörtte birlik kısmı kullanılarak ağın işlem performansı artırmayı hedeflemişlerdir ve farklı alt kümelere bölünmüş eğitim ve test verileri kullanılarak ağa öğretilmeyen 14 farklı saldırı tanımlamışlardır.

Wu ve arkadaşlarının bulunduğu çalışmada [11] Karar Ağaçlarını temel almışlardır. C4.5 karar ağacı yapısı ile diğer bir makine öğrenmesi yöntemi olan Destek Vektör Makinelerinin başarıları kıyaslanmıştır. 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.Karşılaştırma için referans noktası olarak 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.

Folino ve diğ.[12] Genetik programlamayı toplama yöntemi için kullanmışlardır. Dağıtık sistem üzerinde çalışan bir Saldırı Tespit Sistemi önerisi sunmuşlardır. Dağıtık sistemdeki 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 algoritmasınıkullanarak sınıflandırıcılarını birleştirilmektedirler.KDD99,seçilenler arasından %10 verisi ile çalışmışlardır. Kendi algoritmalarını KDD99 yarışmasını birinci ve ikinci bitiren algoritmalar ile karşılaştırmışlardır. Algoritmaları Keskinlik, Recall ve ROC eğrileri olarak daha iyi sonuçlar göstermiştir.

Özgür ve Erdem ' in yapmış olduğu çalışmada[13]

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. Sonuçlar sınıflandırıcılarının eğitim setini nerdeyse ezberlediklerini göstermiş. Fakat her ne kadar ezberleme yapsa da, test veri seti üstünde çok iyi başarı (%99+ ve %91+) göstermişlerdir.

KDD 99 ve NSL-KDD kullanılarak izinsiz giriş tespiti, literatürde iyi çalışılmıştır. Araştırmacılar, test doğruluğunu artırmak için tekli, karma ve topluluk sınıflandırıcıları önerdiler. [15] 'deki Niyaz, iki aşamalı sınıflandırma ile derin öğrenme yaklaşımı (TSL öğrenmiş kendi kendine öğrenme) önermiştir. yaklaşım etiketsiz verilerden iyi özellik temsilini öğrenme ve sınıflandırma için geçerli etiketlenmiş verilere başvurmaktan oluşur. Yazarlar, denetimsiz özellik öğrenimi ve sınıflandırma için soft-maks regresyon için seyrek otomatik kodlayıcı kullanmışlardır. NSL-KDD Veri setini kullanarak 5 sınıfı dikkate alarak % 79.1 test doğruluğu elde ettiler.

Zhang ve Zulkermine rastgele orman algoritması [16] temelli anomali tespit modeli önerdi. Bu hibrit yapı kötüye kullanımı tespiti ve anomali tespitini bulunması konusunda bir birleşim haline geldi.Oluşturdukları bu yapıyı KDD 99 veri setinde değerlendirdiler. Saldırıları 2 sınıfa dönüştürdüler ve % 94,7 saptama oranına sahip oldular. Naive Bayes sınıflandırıcıları [17] 'da izinsiz giriş tespit problemi için de kullanıldı ve rekabetçi sonuçlar verdi. Deneyler, KDD 99 veri seti üzerinde gerçekleştirildi ve 4 sınıf ve 2 sınıf saldırıya odaklandı. Yazarlar, Naive Bayes ile birlikte bir karar ağacı sınıflandırıcısı uyguladı. 4 sınıf vakası için, model karar ağaçları ve Naive Bayes tarafından sırasıyla %91.28 ve %91.47 doğruluk oranı sağlamışlardır. 2 sınıflı dava için karar ağaçları%93.02 sınıflandırma doğruluğu verirken, Naive Bayes algoritmalarında % 91.45 doğruluk .

Javaid ve ark. [18]NSL-KDD veri seti kullanılarak yapılan saldırıların tespitinde seyrek otomatik kodlayıcıların performansını değerlendirmiştir. KDDCUP'99 veri setindeki yüksek doğruluk oranı, [19] 'de, lojistik regresyon softmax ile iki ve dört RBM içeren hibrit bir DBN yapısı kullanılarak rapor edilmiştir.

Lee ve diğerleri [20] 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 [21].

III. VERİ SETİ

KDD-DARPA veri seti Amerikan Hava Kuvvetleri bilgisayar ağına benzer yapıda simülasyon yapılarak oluşturulmuştur.. DARPA tarafından sponsor edilen çalışma MIT Lincoln Labs tarafından 1998 yılında tamamlanmıştır [6]. Bir yıl sonra aynı çalışma Bilgisayar Güvenliği ile çalışanlardan alınan yorumlar ile iyileştirmeler yapılarak güncellenmiştir ve KDD99 veri seti ortaya koyulmuştur.

Veri seti 494021 adet veriden oluşmaktadır. İçerisinde 41 adet özellik barındırmaktadır. Bu özelliklerin bir kısmı kategorik bir kısmı ise sayısal değerler içerir [7].
Hedef özelliğimiz ise atak tiplerini veya normal durumu simgeler. 22 farklı atak çeşidi mevcuttur. 1 adet de normal durum sözkonusudur. Toplamda 23 adet sınıfa ayrılabilir [7].

A. Saldırı Tipleri

Dos Attacks

1) *Back Dos:* Back DOS saldırısı bir Apache Web sunucusunu hedefleyen saldırıdır. Bu saldırı sırasında bir saldırgan birçok Forward Slash("/") içeren URL'lerle istek gönderir, sunucu bu uygulamaları işlemeye çalıştığında yavaşlar ve diğer istekleri işleyemez.
2) *Neptune Dos:* Neptune Dos saldırısı bir Syn Flood saldırısıdır ve hizmet reddi üretmeyi amaçlar. Bu saldırı, hedef makineye bir dizi SYN talebi gönderilerek TCP bağlamında gerçekleştirilir. Bir SYN taşmasının genel çalışması, bir istemci ile bir sunucu arasında bir TCP bağlantısı başlatıldığında ve bir mesajlama gerçekleşeceği zaman başlar. SYN saldırısı, var olmayan veya geçersiz bir kaynağı olan bir IP adresine sahip ana bilgisayara çok sayıda SYN isteği gönderiyor. SYN'e duyarlı sistemler bağlantıları açar ve bir ACK paketi almayı bekliyorlar. Bu sayıdaki SYN paketlerini çok fazla alarak ve hedef makine tarafından sıralı istekleri saklamak için kullanılan kaynaklar tükenirse, hedef makine bir çökmeye veya sistemin yeniden başlatılmasına neden olabilir.
3) *Teardrop Dos:* Teardrop saldırısı, Win32 sistemlerini ve Linux sistemlerini hedef alan bir saldırıdır. Bu saldırı örtüşen UDP paketleri gönderilmesine izin veriyor. Saldırılan sistemi paketi aldığında, yeniden inşa etmeye çalışır ve bunu yapamaz, böylece koruya ve sistemin çökmesine neden olur.
4) *Smurf Dos:* Smurf saldırısı, ağları hedef alan DoS saldırıları ailesine ait bir saldırıdır. ICMP (Internet Control Message Protocol) veya UDP (User Datagram Protocol) üzerinde bir Smurf saldırı gerçekleştirilebilir. Boyle bir saldırı yapmak için, saldırgan bir yayın sunucusuna (broadcast server) bir ICMP paketi göndererek başlar ve sunucu bu paketi ağındaki tüm makinelere iletir. Makineler ICMP paketini aldıklarında, ilk ICMP paketini gönderen makineye ikinci bir ICMP paketi ile cevap vereceklerdir. Eğer saldırgan sahte sunucu IP adresine ilk ICMP paketini yayın sunucusuna yolladıysa, hedef makinelerin cevapları bu IP adresine yanlış olacaktır. Bu nedenle bant genişliği kaybı ve yavaşlama ve kurban makineleri sisteminin tıkanması söz konusu olacaktır.
5) *POD Dos:* POD saldırısı Ping Of Death anlamına geliyor ve en eski ağ saldırılarından biri olan DoS saldırısıdır. Bu işlem, 65536 bayt olan izin verilen maksimum boyutu aşan toplam boyutta bir IP datagramı oluşturarak yapılır. Paket, bir TCP/IP yığını dahili olan bir sisteme gönderildiğinde, makinenin çökmesine neden olur.

6) *Land Dos:* Land Dos, Win32 sistemlerini hedef alan hizmet reddi saldırılarının bir parçası olan 1997'den itibaren ağ odaklı bir saldırıdır. Bu saldırı, sahte bir paket gönderir ve istenmeyen bir şekilde davranmasını sağlamak için sistemle tutarsız tutar. Bu saldırı bazı TCP / IP uygulamalarında bir güvenlik açığından yararlanmak için sahtekarlık tekniğini kullanır. Kaynak ve hedef IP paket alanlarında aynı IP adresine ve aynı port numarasına sahip bir paket gönderilmesine izin verir. Sonradan, hedef makinenin sistemini çökertir ve dengesiz bir duruma getirir.

U2R Attacks

1) *Bufferoverflow attack:* Bufferoverflow etkili ve aynı anda gerçekleştirmek için çok karmaşık bir saldırıdır. Saldırının çalışma prensibi, programa beklenenden daha fazla veri göndererek bir programda isteğe bağli bir kod çalıştırmak için arabellek taşmasıdır. Bu, bir çökmeye ve sisteme erişim sağlayan saldırılan makinesini kilitleme sistemine neden olur. Bir Bufferoverflow saldırısının uygulanması, mimari programlar ve işlemciler hakkında bilgi sahibi olmasını gerekir.
2) *Loadmodule attack:* Bu saldırı, Xnews sistemini kullanan SunOS 4.1 sistemlerine karşıdır. Loadmodule Program, Xnews sistemi sunucusu tarafından, her iki sürücüyü de çalışan sistemde dinamik olarak yüklenebilen çekirdeği yüklemek için bu modülleri kullanmak üzere dizinde (/ dev) özel cihazlar oluşturmak için kullanılır. Programın yük modülünün ortamını sterilize etme şeklindeki bir hata nedeniyle, yetkisiz kullanıcılar yerel makineye kök erişimi sağlayabilir.
3) *Perl attack:* Perl, metin dosyalarının çıkarılması ve veri raporlama için optimize edilmiş bir dildir. Önceden, metinlerin işlenmesi kabuk gibi komutları kullanarak (sed, awk, cut ve grep expr) komutları kullanılarak yapılır. Bu tür bir emrin kullanılması, verilerin bir işlemden diğerine taşınmasının zorluğu, bir komutun belirli bir uygulamasına bağlı olma, birkaç programın başlatılması nedeniyle yavaş ve giriş verileri biçiminin esnek olmama gibi sorunlara yol açar. Perl sözdizimi, C dilinde sözdizimi ve sistem kütüphanesi işlevlerinin bir kısmını ekleyen tüm mini dillerini toplar ve ödünç alır. Slowloris, Perl'de yazılmış örnek bir komut dosyasıdır; bu komut dosyası, bir makinenin bir web sunucusunu minimum bant genişliği kullanarak ve rapor vermeden servisler ve portlar üzerinde son derece etkili Slowloris, DoS kullanıyor, komut dosyası özellikle ağdaki sunucuların yarısından fazlasını temsil eden Apache 1.x ve Apache 2.x sunucularını etkiler.
4) *Rootkit attack:* Bir rootkit, izinsiz girdikten ve arka kapı taktıktan sonra değişiklikleri gizlemek için kullanılır. Rootkit kolayca tespit edilemez, varlığını ayrıntılı bir analizle ortaya çıkartabilir. Kamuflaj rootkit'in temel rolü, bilgisayar korsanlarının, ilk erişime izin veren kusuru tekrar kullanmadan sistemin kalbini hedef almalarını sağlayan bir veya daha fazla backdoor geliştirilmesidir. Rootkit saldırısı sistem komutlarında ve hatta çekirdek düzeyinde değişiklikler yapabilir. Ama tek başına büyük sorunlara neden olamaz çünkü amacı tehlikeli olabilecek bir programı gizlemektir, bu nedenle sık sık diğer programlarla birleştirilir. Genel olarak, bu tür saldırılar dört tür program kullanır: Trojanlar, backdoor, ağ dinleyicileri ve log dosyalarının temizleyicileri.

R2L Attacks

1) *FTP-write attack:* Saldırı ftp-write, saldırganın, PORT komutunu kullanarak, saldırılan makinesini istek için Middle Man olarak kullanarak, portlara dolaylı yoldan erişim talebinde bulunabileceği bir FTP protokolüdür. Bu saldırı, tarama ana bilgisayarlarını gizli bir şekilde yerleştirmek, doğrudan bir bağlantı yoluyla erişimin mümkün olduğu bireysel ve özel bağlantı noktalarına erişmek için kullanılabilir. Bu saldırı son derece etkili bir saldırıydı, ancak artık alakalı değil. Saldırı bir IP adresi sahtekarlığı durumuydu.
2) *Guess-password attack:* Bu saldırı bir bilgisayar sistemini kırmak için yaygın bir tekniktir. İnternetin evrimi, kişisel bilgi sızıntılarını çalmak için çeşitli fırsatlar yarattı. Tahmin-şifre saldırılarının farklı amaçları olabilir, ancak bir sistemin veya bir bilgisayar ağının kontrolünü almak için aynı hedefe sahipler.
3) *Multihop attack:* Multihop saldırı, yerel Remote, yani uzak bir makineye erişim izni olmadan erişime yönelik bir saldırı türüdür. Multihop olmadan, kablosuz ağlar kimliğe bürünme nedeniyle saldırılara karşı özellikle savunmasızdır. Bu tür bir saldırıda, düşman bir rakip kimliğini değiştiren düğümleri kontrol eder.
4) *Warezmaster attack:* Warezmaster Remote2Local bir saldırıdır ve dosya aktarım protokolü (FTP) sunucusuyla ilişkili bir anormallik sisteminden yararlanmak için kullanılır. Genel olarak konuk kullanıcılar hiçbir zaman bir FTP sunucusuna yazma hakkına sahip değildir ve bu nedenle sunucuya hiçbir zaman dosya gönderemezler. Herkes müşteri hesaplarını kullanarak bir FTP sunucusuna bağlanabilir. Bu saldırı, bir FTP sunucusu sistem kullanıcıları için yanlışlıkla yazma hakları sağladığında ortaya çıkabilir ve bu nedenle herhangi bir kullanıcı giriş yapabilir ve dosya gönderebilir.
5) *Warezclient attack:* Warezclient saldırısı, bir warezmaster saldırısı yapıldıktan sonra bir FTP bağlantısı sırasında herhangi bir yasal kullanıcı tarafından başlatılabilir. Bir warezclient saldırısı sırasında, kullanıcılar daha önce başarılı bir warezmaster saldırısı ile gönderilen yasa dışı yazılımları indirirler. Bu işlem FTP sunucusundan dosya indirmeyi gerektirdiğinden, bu dinamik saldırı tamamen yasal bir işlem gibi görünüyor.
6) *Imap attack:* Bu saldırı, IMAP sunucusu Redhat Linux 4.2'de bir önbellek akışından yararlanıyor. Bu, uzak saldırganların kök ayrıcalıklarıyla istediği komutlar çalıştırmasına izin verir. Imap-attack, Linux platformu için önceden derlenmiş ikili dosyaları içeren Impack

1.03 saldırısının araç kutusunun bir parçasıydı. Bu programlar, önbellek akışından yararlanmak ve bir kök kabuğuna erişmek için gereken mesajı göndermenizi sağlar. Impack, bu programların nasıl kullanılacağına dair ayrıntılı talimatlar içerir, bu çok tehlikeli bir saldırıya neden olur çünkü Linux makinesi olan herhangi bir kullanıcı Imap saldırı talimatlarını izleyerek gerçekleştirebilir.

7) *Phf attack:* Saldırı, Phf'nin http sunucu ayrıcalıklarında komutları çalıştırmak için kusurlu yazılmış bir CGI komut dosyasını kötüye kullanmasına izin veriyor. CGI programı CGI fonksiyonuna ne olursa olsun "escape shell cmd ()" kabuktan kütüphane çağrılarını yasaklamak için bu saldırıya açık olabilir. Bu kusur veya güvenlik açığı, Apache Web sunucusu için örnek kodla birlikte dağıtılan "Phf" programına yansır.
8) *Spy attack:* SpyAttack (casus yazılım), bir programın ana koduna entegre edilmiş küçük bir kod kümesi aracılığıyla yapılan bir saldırıdır. Saldırıda kullanılan bu küçük kod parçaları genellikle kullanıcılar tarafından İnternet üzerinden indirilen programlarda bulunur. Casus saldırılarında kullanılan kodların güvenlik sorunlarına neden olması amaçlanmamıştır, ancak İnternet kullanıcılarının özel yaşamları ve alışkanlıkları hakkında bilgi sahibi olmayı amaçlamaktadır.

Probe Attacks

1) *Ipsweep attack:* Ipsweep saldırısı, hangi ana makinenin ağda olduğunu belirlemek için yapılan bir analiz ve izleme olarak düşünülebilir. Bu bilgi saldırganlar için çok önemlidir çünkü aşama aşama saldırılarında ve savunmasız makinelerin aranmasında faydalıdır.
2) *Nmap attack:* Network Mapper olarak da bilinen Nmap, ağ keşfi ve güvenlik denetimi için ücretsiz ve açık kaynaklı bir yardımcı programdır. Güvenlik duvarı tanımlamasını sağlayan araçlar ve teknikler çok çeşitlidir. Bu araçlar, bilgisayar korsanları tarafından güvenlik duvarının çalışmasını ve ağ efektini kontrol etmek için kullanılır. İlk önce, saldırgan güvenlik duvarını bulmaya çalışır, ardından güvenlik duvarındaki kusurdan yararlanmayı veya paket filtresindeki bir kusuru tespit etmek için güvenlik duvarının diğer kurallarını aramaya çalışır. Bu kuralları tanımlamak için Nmap gibi bir port taraması kullanılmalıdır.
3) *Satan attack:* Satan (Ağları Denetleme için Güvenlik Analiz Aracı) aynı zamanda ağa erişimi olan herkes için mevcut olan bilgileri toplayan bir probe saldırısıdır. Uzaktaki bir ana bilgisayarı veya ana bilgisayar kümesini inceler ve NIS, parmak, NFS, ftp ve tftp, rexd ve diğer hizmetleri uzaktan denetleyerek mümkün olduğunca fazla bilgi toplar. Bu bilgiler, genellikle yanlış ayarlanmış veya yapılandırılmış ağ servisleri, sistem veya ağ yardımcı programlarında iyi bilinen hatalar veya kötü veya cahil politika kararları şeklinde, çeşitli ağ bilgi servislerinin yanı sıra potansiyel güvenlik kusurlarının varlığını içerir. Daha sonra bu verileri raporlayabilir veya olası güvenlik sorunlarını daha fazla araştırabilir.
4) *Portsweep attack:* Bilgisayarda Portsweep, bir ağ sunucusundaki açık bağlantı noktalarını bulmak için kullanılan bir tekniktir. Bu teknik, bilgisayar sistemlerinin yöneticileri tarafından ağlarının sunucu güvenliğini kontrol etmek için kullanılır. Fakat aynı teknik bilgisayar korsanları tarafından bilgisayar sistemindeki kusurları bulmak için de kullanılır. Üçüncü taraf bir sistemde port taraması yapmak genellikle izinsiz giriş denemesi olarak kabul edilir. Algılama sistemlerinin ve güvenlik duvarlarının dikkatinden kaçmak için, taramalar rasgele sırayla, birkaç gün içinde bile çok yavaş bir hızda veya birden fazla IP adresinden yapılabilir. Portsweep saldırısı genellikle TCP ve UDP'de yapılır.

IV. MODEL

A. Veri Ön İşleme

Günümüz dünyasında çok büyük ölçekli veriler kullanılmaktadır.Bu veriler gürültülü,eksik veya tutarsız olabilmektedir.Kullanılan bu verilerden doğru bir şekilde analiz yapmak ve bunlara dayanarak sonuç çıkarmak çok zor bir hal almaktadır.Bu yüzden verinin kalitesi önemli bir unsurdur.[4]Verinin kalitesinin artırmak içinde veri ön işleme yapılmaktadır.

Çok sayıda veri ön işleme tekniği bulunmaktadır. Bunlardan biri olan veri temizleme (data cleaning) verilerdeki gürültünün giderilmesi ve tutarsızlıkların düzeltilmesi için uygulanmaktadır. Bir diğer teknik olan veri birleştirme (data integration) ise farklı kaynaklı verileri uygun bir veri tabanında birleştirmektedir. Normalleştirme gibi veri dönüştürme yöntemleri (data transformations) uygulanmaktadır. Veri indirgeme (data reduction) de ise fazla olan bazı değişkenlerin atılması ve birleştirilmesi veya kümeleme yolu ile veri büyüklüğünün azaltılması amaçlanmaktadır. Bu sözü edilen veri ön işleme teknikleri, veri madenciliğinden önce uygulanarak elde edilen sonuçların kalitesi ve veri madenciliği için harcanacak zaman artırılmış olmaktadır[2].

Biz bu projemizde veri önişleme adımı olarak veri dönüştürme aşamasını gerçekleştirdik.Veri setimizde bulunan kategorik verileri modelimizde kullanamadığımız için verileri sayısal değerlere dönüştürdük.Bunun içinde sklearn isimli kütüphanede ki LabelEncoder sınıfından yararlandık. Label Encoder 0 ile nsınıf-1 arasında bir değere sahip etiketleri kodlar; burada n, farklı etiketlerin sayısıdır. Bir etiket tekrarlarsa, daha önce atanan ile aynı değeri atar.[5]

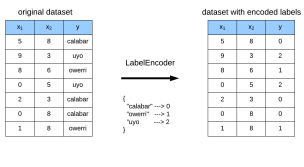

Fig. 1. Kategorik Verileri Sayısala Dönüştürme

Biz de bu projemizde veri ön işleme aşamalarından bir olan veri boyutunun azaltılması işleminden yararlanılmaktadır.

Veri boyutunun azaltılması kısaca, büyük veri kümelerinin depolanması ve analiz edilmesinde karşılaşılan sorunları aşmak için veri kümesinden gereksiz değişkenlerin çıkartılması olarak tanımlanmaktadır. Biz de veri boyutumuzu azaltılmak için kullanılan yöntemlerin başında gelen özellik seçimi kullanılmaktadır.Amacımız en iyi özellikleri belirleyip veri boyutumuzu düşürmektedir.[1]

B. Özellik Seçimi

Özellik seçiminde ki amaç veri setini en iyi şekilde temsil edecek alt kümeyi seçmektedir.Özellik seçimi genel olarak doğruluk ve ölçeklenebilirlik için kullanılmaktadır. [3]Veri setindeki bazı özellikler işlem performansını olumsuz etkileyecek gürültüye sahip olduğundan bu özelliklerin veri kümesi içinden silinmesi, işlem sonucunun doğruluğunun artmasında etkili olabilmektedir.

Özellik seçiminin genel adımlarına bakacak olursak,öncelikle veri setinden bir alt özellik kümesi oluşturulmakta ve daha sonra ele alınan özellikler için seçilen özellik seçme yöntemi ile bir değerlendirme yapılarak ilgili özelliğin seçilip seçilmeyeceğine karar verilmektedir. Seçilmesine karar verilen özellik ilgili alt kümeye dahil edilmekte ve algoritmanın durdurma kriteri sağlana kadar adımlar devam etmektedir.[1]

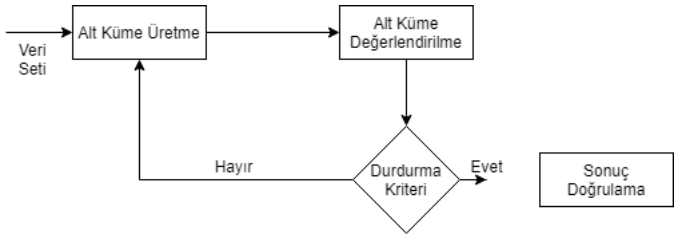

Fig. 2. Özellik Seçimi Şeması

Özellik seçiminde kullanılan yöntemler, sadece istatistiksel bilgiye dayalı olan filtreleme (filter) yöntemleri, özellikler üzerinde arama işlemleri gerçekleştiren sarmal (wrapper) yöntemler ve en iyi bölen ölçütünü bulmaya dayalı olan gömülü (embedded) yöntemler olmak üzere genel olarak üç grupta toplanmaktadır.[1]

Proje de özellik seçimi yöntemlerinden biri olan filtreleme yöntemi başlığı altında geçen bilgi kazancı(Entropy) yöntemi kullanılmıştır.

Filtreleme Yöntemleri

Filtreleme yöntemleri en eski özellik seçim yöntemleri olarak bilinmektedir.Bu yöntemlerde, veri kümesinde bulunan her bir özellik için değerlendirme fonksiyona sokarak bir değer hesaplanmakta ve hesaplanan bu değerler içerisinde en yüksek değerlere sahip olan özellikler en iyi özellik alt kümesine,Korelasyon tabanlı özellik seçimi seçilmektedir.Filtreleme yöntemlerinde Fisher Skor,t-Skor,Korelasyon tabanlı özellik seçimi,Bilgi Kazancı vb. bir çok yöntem bulunmaktadır.[1]

Bilgi Kazancı(Entropi) Bilgi kazancı entropi dayalı bir özellik seçim yöntemidir.Entropy bir sistemdeki düzensizliğin ölçüsüdürEntropi 0 ve 1 aralığında değerler alır ve 1 değerine yaklaştıkça belirsizlik artar. Yüksek entropiye sahip veri daha çok bilgi içerir. Y özelliğini tanımak için gereken bilgi ile X özelliği de kullanılarak Y özelliğini tanımak için gereken bilgi farkını gösteren Bilgi Kazancı değerini aşağıdaki gibi hesaplanmaktadır[3]

$$\mathbf{H}(\mathbf{Y}) = -\sum_{y \epsilon Y} p(y) \log_2(p(y)) \tag{1}$$

$$\mathbf{H}(\mathbf{Y}/\mathbf{X}) = -\sum_{x \epsilon X} p(x) \sum_{y \epsilon Y} p(y/x) \log_2(p(y/x)) \tag{2}$$

$$BilgiKazancı = H(Y) - H(Y/X) \tag{3}$$

Projemizde entropi eşik değeri olarak 0.03 değerinden büyük olan özellikler seçilmiştir.Bu yöntem ile 41 tane olan özellik sayımızı 11'e düşürmüş olduk.Bu durum hem işlemci gücümüzü fazla kullanmamamızı hem de daha iyi bir isabetle tahmin yapabilmemizi sağlamıştır.

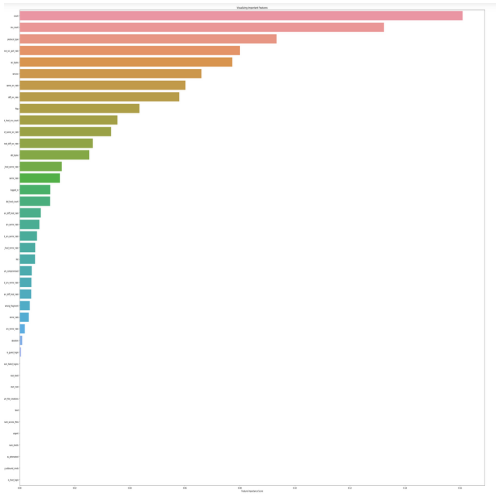

Fig. 3. Entropi Grafiği

C. Modelin Yapısı

Model Çok Katmanlı Sinir Ağı(MLP) yapısıyla oluşturulmuştur. 1 Giriş katmanı 1 gizli katman ve 1 çıkış katmanından oluşmaktadır. Tam Bağlantılı Ağ(FUlly Connected) olarak oluşturulmuştur. Giriş katmanındaki nöron sayısı Özellik Seçimi başlığında anlatın elemeler sonucunda dinamik olarak seçilmektedir. Elimizdeki verilerden elde ettiğimiz elemeden sonra 11 özellik seçilmiş ve kullanılmıştır. Giriş katmanındaki nöron sayısı özellik sayısına eşittir. Gizli katmanda 512 nöron bulunmaktadır. Aktivasyon fonksiyonu olarak TanH seçilmiştir. Buradan elde edilen 512 veri çıkış katmanına aktarılmaktadır. Çıkış katmanında Softmax kullanılmıştır. Bu aktivasyon fonksiyonu sayesinde her bir kategorinin olma olasılığı elde edilmektedir. En yüksek olasılık seçilir. Bu seçim modelimizin çıktısı olmaktadır.

D. Olasılıksal Dereceli Azalma(SGD)

Stochastic Gradient Descent (SGD), (lineer) Destek Vektör Makineleri ve Lojistik Regresyon gibi konveks kayıp fonksiyonları altında lineer sınıflandırıcıların ayırt edici öğreniminde basit ama çok etkili bir yaklaşımdır. SGD, makine öğrenmesinde uzun zamandır bulunsa da , son zamanlarda büyük ölçekli öğrenme bağlamında dikkat çekmektedir.[14]

SGD, metin sınıflandırmasında ve doğal dil işlemede sıklıkla karşılaşılan büyük ölçekli ve seyreltilmiş makine öğrenmesi problemlerine başarıyla uygulanmaktadır. Verilerin seyrek olduğu göz önüne alındığında, bu modüldeki sınıflandırıcılar, 10^5 den fazla eğitim örneği ve 10^5'den fazla özellik içeren problemlere kolayca ölçeklenebilir. [23] SGD'nin avantajları ve dezavantajları bulunmaktadır. Avantajları:

- Verimlilik
- Uygulama kolaylığıdır

Dezavantajları:

- SGD, düzenlileştirme parametresi ve yineleme sayısı gibi bir dizi hiper parametre gerektirir
- SGD özellik ölçeklemeye duyarlıdır

Gradient Descent'te w_j değerini her bir adımda güncellenebilmek için tüm verisetini gezmesi gerekmektedir. Bütün veri setini gezmekten kasıt $\sum$ notasyonu ile yapılan işlemdir. SGD'te ise amaç her bir adımda verisetindeki her bir örneği gezdikten sonra güncellemektir.

SGD'nin en büyük avantajı, eğitim örneği sayısı ve verimliliğinin genellikle doğrusal olmasıdır. X, bir boyut matrisi (n, p) ise, k , yineleme sayısı (dönemler) ve $\bar{p}$, örnek başına sıfıra yakın niteliklerin ortalama sayısı olan $O(kn\bar{p})$ maliyetine sahiptir. Bununla birlikte, son teorik sonuçlar, eğitim seti boyutu arttıkça, istenen optimizasyon hassasiyetini elde etmek için çalışma süresinin artmadığını göstermektedir.

Her iterasyondaki parametrelere bir sonraki güncellemeyi hesaplamak için tam eğitim setini kullanan sınırlı bellek BFGS gibi toplu yöntemler, yerel optimuma çok yakınsaklık kazanma eğilimindedir. Ayrıca, ayar yapmak için çok az sayıda hiper parametreye sahip oldukları için, iyi bir raf uygulamasına (örn., MinFunc) sağlanan bir çalışma ortamı sağlamak için açıktırlar. Bununla birlikte, pratikte, veri seti ana belleğe sığmayacak kadar büyükse, tüm eğitim seti için maliyet ve gradyan hesaplama, tek bir makinede çok yavaş ve bazen zor olabilir. Toplu optimizasyon yöntemleriyle ilgili bir diğer konu da, yeni verileri "çevrimiçi" bir ayara dahil etmek için kolay bir yol sağlamamalarıdır. Stochastic Gradient Descent (SGD), yalnızca bir veya birkaç eğitim örneğini gördükten sonra objektifin negatif eğimini izleyerek bu konuların her ikisine de değinir. SGD'nin kullanımı Sinir ağı ayarında, tam eğitim seti üzerinden geri yayılımı çalıştırmanın yüksek maliyeti tercih edilir. SGD bu maliyetten kurtulabilir ve yakınsama hızlanır.[14]

Standart dereceli azalma algoritması objektif ve J() parametrelerini şu şekilde güncellenir:

$$\theta = \theta - \alpha \Delta_\theta E[J(\theta)] \tag{4}$$

Burada, yukarıdaki denklemdeki beklenti, tam eğitim seti üzerinden maliyet ve dereceyi değerlendirerek yaklaşıklaştırılır. Stochastic Gradient Descent (SGD), güncellemedeki beklentiyi ortadan kaldırır ve yalnızca bir veya birkaç eğitim örneği kullanılarak parametrelerin değişim derecesini hesaplar. Yeni güncelleme,

$$\theta = \theta - \alpha \Delta_\theta J(\theta; x^{(i)}, y^{(i)}) \tag{5}$$

eğitim setindeki $x(i), y(i)$ ile eşleştirilmesi.

Genellikle, SGD'deki her parametre güncellemesi, tek bir örneğe kıyasla birkaç eğitim örneği veya bir minibatch olarak hesaplanır. Bunun nedeni iki yönlüdür: önce parametre güncellemesindeki değişimi azaltılır ve daha istikrarlı yakınsama getirilir, ikincisi, hesaplamanın maliyetinin ve derecenin iyi vektörize edilmiş bir hesaplamada kullanılması gereken yüksek düzeyde optimize edilmiş matris işlemlerinden yararlanmasını sağlar. Minibatch'in en uygun boyutu farklı uygulamalar ve mimariler için değişiklik gösterse de tipik bir minibatch boyutu 256'dır.[24,25]

SGD'de, öğrenme hızı genellikle karşılık gelen dereceli azalmayla ilgili öğrenme oranından çok daha küçüktür, çünkü güncellemede çok daha fazla varyans vardır. Doğru öğrenme oranını ve zamanlamasını seçme (örneğin, öğrenme ilerledikçe öğrenme oranının değerini değiştirme) oldukça zor olabilir. Uygulamada iyi çalışan bir standart yöntem, ilk döngüde(initial epoch) (eğitim seti üzerinden tam geçiş) veya iki eğitimde sabit yakınsaklık sağlayan küçük bir sabit öğrenme oranını kullanmak ve yakınsaklık yavaşlarken öğrenme oranının değerini yarıya indirmektir. Daha iyi bir yaklaşım, her döngüden sonra saklanan bir kümeyi değerlendirmek ve döngüler arasındaki hedef değişikliği küçük bir eşiğin altında olduğunda öğrenme hızını ayarlamaktır. Bu, yerel bir optimuma iyi yakınsama eğilimi gösterir. Yaygın olarak kullanılan diğer bir zamanlama, her yinelemede t öğrenme hızını a / (b + t) olarak hesaplamaktır; burada a ve b ilk öğrenme hızını belirtir ve döngü sırasıyla başlatılır. Daha sofistike yöntemler, en iyi güncellemeyi bulmak için bir geri çekilme hattı aramasını kullanmayı içerir.

SGD ile ilgili son ama önemli nokta, veriyi algoritmaya sunmamızın sırasındadır. Veriler anlamlı bir sırayla verilirse, bu dereceyi bozabilir ve zayıf yakınsama yol açabilir. Genellikle bunu önlemek için iyi bir yöntem olan, verilerin her bir eğitim aşamasından önce rastgele karıştırılmasını sağlamaktır.

$$\nu = \gamma\nu - \alpha \Delta_\theta J(\theta; x^{(i)}, y^{(i)})\theta = \theta - \nu \tag{6}$$

E. Aktivasyon Fonksiyonları

Belirli bir eşik değerine göre nöronların aktif olup olmamalarını sağlayan fonksiyonları içerir. Aktivasyon işlemi, bir öznitelik haritasının her bileşenine uygulanan doğrusal olmayan aktivasyon fonksiyonuyla lineer bir filtre izlenerek elde edilir.

Softmax Fonksiyonu.Sigmoid fonksiyonuna çok benzer bir yapıya sahiptir. Aynı Sigmoid'te olduğu gibi sınıflayıcı olarak kullanıldığında oldukça iyi bir performans sergiler. En önemli farkı sigmoid fonksiyonu gibi ikiden fazla sınıflamak

gereken durumlarda özellikle derin öğrenme modellerinin çıkış katmanında tercih edilmektedir. Girdinin belirli sınıfa ait olma olasılığını 0–1 aralığında değerler üreterek belirlenmesini sağlamaktadır. Yani olasılıksal bir yorumlama gerçekleştirir.

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))} \tag{7}$$

Tanh Fonksiyonu [-1,1] aralığında çıktı üreten doğrusal olmayan bir fonksiyondur.

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{8}$$

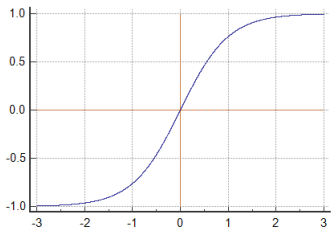

Fig. 4. Hyperbolic Tangent

F. RMSProp

Sinir Ağlarını, uzaklık ölçütlerine ve Gauss aktivasyon fonksiyonlarına dayanarak sadece mini-batch gradyan iniş veya momentum kullanarak eğitmek mümkün değildir. Bu yöntemle yakınsama başarılamaz ve ya çok yavaş bir yakınsama elde yapılabilir. Bu yüzeysel ağlar için bile geçerlidir. Çözüm, ilk etapta eğitimi mümkün kılan RMSProp'u(Kök Ortalama Kare Yayılımı) uygulamaktadır. [22]

RMSProp'ta, öğrenme hızı $\forall j \in 1, ..., sl$ ve $\forall l \in 1, ..., L$ koşulları altında ${c_j}^{(l)}$ ve ${r_j}^{(l)}$ vektörlerinin her birine uygulanır. Buradaki fikir, her ağırlık için bitişik mini-batch üzerinden karesel gradyanların hareketli ortalamasını tutmaktır:

$$\bar{v}({c_j}^{(l)}, t) = \gamma \bar{v}({c_j}^{(l)}, t-1) + (1-\gamma)(\partial J / \partial {c_j}^{(l)})^2 \tag{9}$$

$$\bar{v}({r_j}^{(l)}, t) = \gamma \bar{v}({r_j}^{(l)}, t-1) + (1-\gamma)(\partial J / \partial {r_j}^{(l)})^2 \tag{10}$$

γ , unutkanlık faktörü olduğu zaman tipik olarak 0.9'dur. Daha sonra bir ağırlık için öğrenme oranı, bu hareketli ortalamaya bölünür:

$${c_j}^{(l)} := {c_j}^{(l)} - \frac{\eta}{\sqrt{\bar{v}({c_j}^{(l)}, t)}} . \partial J / \partial {c_j}^{(l)} \tag{11}$$

$${r_j}^{(l)} := {r_j}^{(l)} - \frac{\eta}{\sqrt{\bar{v}({r_j}^{(l)}, t)}} . \partial J / \partial {r_j}^{(l)} \tag{12}$$

RMSProp, farklı uygulamalarda öğrenme hızının mükemmel bir şekilde uyarlandığını göstermiştir.

V. Sonuçlar

Modelimizde Çok Katmanlı Sinir Ağı (MLP) kullanılmıştır. Oluşturulan modelde 99.06% Doğruluk(Accuracy) 99.01% Duyarlılık(Recall) 98.26%Hassasiyet (Precision) ve 98.62% F puanı elde edilmiştir.

Doğruluk(Accuracy)	99.06%
Duyarlılık(Recall)	99.01%
Hassasiyet(Precision)	98.26%
F-puanı	98.62%

Modelin test sonucunda elde ettiği doğruluk eğrisi Şekil 5 de görülmektedir.

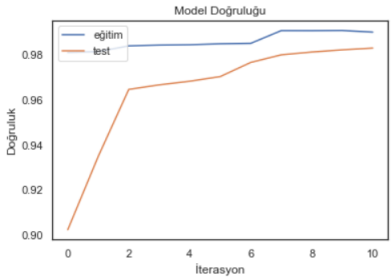

Fig. 5. Doğruluk Grafiği

Modelin Kayıp(loss) grafiği şekil 6 de görülmektedir. Testin sonunda 3.24% kayıp elde etmiştir.

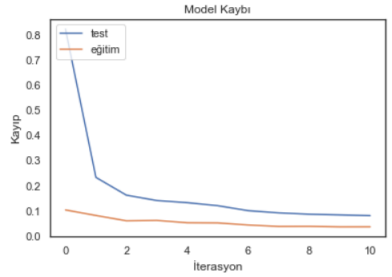

Fig. 6. Kayıp Grafiği

Tarih	Makale Adı	Algoritmalar	Doğruluk	Duyarlılık	Hassasiyet	F skoru
15 Şubat 2011	ZEKİ SALDIRI TESPİT SİSTEMİ TASARIMI ve GERÇEKLEŞTİRİLMESİ	MLP (KDD özellik vektörü) MLP (IP Protokolü tabanlı)	99 99	-	-	-
Mayıs 2012	Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması	Karar Ağaçları Yapay Sinir Ağları SVM AdaBoost	91.7406 92.4821 92.3049 91.3886	-	-	-
Temmuz 2017	Network IntrusionDetectionforCyber Security using UnsupervisedDeep Learning Approaches	K-Means US-ELM AE+K-means RBM+K-means	%87.72 %89.17 %90.86 %90.86	-	-	-
Ekim 2017	A New Deep Learning Approach for Anomaly Base IDS using Memetic Classifier	Deep learning Memetic GA Native Bayes Decision Tree	-	-	98.11 97.22 98.34 97.68 99.50	-
Ekim 2017	A Few-shot Deep Learning Approach for Improved Intrusion Detection	SVM k-NN Deeplearning	95.27 96.19 94.62	-	-	-
2018	Fast Activation Function Approach for Deep Learning Based Online AnomalyIntrusion Detection	SSAELM DBN	- -	95.2 98.2	95.28 98.58	94.00 97.00
Şubat 2018	A Deep Learning Approach to Network Intrusion Detection	DBN S-NDAE	99.49 99.49	99.49 99.49	94.51 100.00	96.94 99.75
28 Ağustos 2018	An IntrusionDetectionSystem Using a Deep Neural Network WithGatedRecurrentUnits	BGRU+MLP GRU+MLP BLSTM+MLP LSTM+MLP GRU LSTM MLP	99.84 99.28 98.57 98.51 92.28 91.91 91.88	-	-	-

TABLE I
LITERATÜRDEKI ÇALIŞMALAR

REFERENCES

[1] H.Budak "Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım" Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi , Cilt 22, Özel Sayı, 21-31, 2018

[2] HAN, J. and M. KAMBER (2001),"Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers", USA, 550p.

[3] B.Yazıcı, F.Yaslı,H.Gürleyik, U.Turgut,M.Aktas,O.Kalıpsız"Veri Madenciliğinde Özellik Seçim Tekniklerinin Bankacılık Verisine Uygulanması Üzerine Araştırma ve Karşılaştırmalı Uygulama" Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul

[4] A.Oğuzlar, "Veri Ön İşleme",Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Sayı: 21, Temmuz-Aralık 2003, ss. 67-76.

[5] https://www.kaggle.com/questions-and-answers/61046.

[6] 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.

[7] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

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

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

[10] Gazi Üniv. Müh. Mim. Fak. Der. J. Fac. Eng. Arch. Gazi Univ. Cilt 26, No 2, 325-340, 2011

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

[12] . 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.

[13] .Özgür H.Erdem BİLİŞİM TEKNOLOJİLERİ DERGİSİ, CİLT: 5, SAYI: 2, MAYIS 2012

[14] AL: A fast DNN optimization method based on curvature information-ArXiv2019 Maximus MutschlerAndreas Zellt

[15] . Niyaz, W. Sun, A. Y. Javaid, and M. Alam, "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), BICT-15, vol. 15, 2015, pp. 21–26.

[16] . Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. IEEE, 2006, pp. 8–pp.

[17] . B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," in Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004, pp. 420–424.

[18] R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Effective discriminantfunction for intrusion detection using SVM," inProc. Int. Conf. Adv.Comput., Commun. Inform. (ICACCI), Sep. 2016, pp. 1148–1153.

[19] . Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection sys-tem based on KNN classification algorithm in wireless sensor network,"J. Elect. Comput. Eng., vol. 2014, Jun. 2014, Art. no. 240217.

[20] .-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.

[21] . 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].

[22] INT++: Robust Visual Tracking via Adversarial Positive Instance Generation2018 IEEE/CVF Conference On Computer Vision And Pattern Recognition2018 Xiao WangChenglong LiBin LuoJin Tang

[23] euper (Fehr, Janis & Pfreundt, Franz-Josef. (2015). Asynchronous Parallel Stochastic Gradient Descent - A Numeric Core for Scalable Distributed Machine Learning Algorithms.

[24] Local Block Coordinate Descent Algorithm for the Convolutional Sparse Coding ModelArXiv2018 Ev ZisselmanJeremias SulamMichael Elad

[25] New Type of a Wavelet Neural NetworkOptical Memory And Neural Networks2018 Alexander EfitorovS. A. Dolenko