AĞ SALDIRI TESPİTİNDE SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

MUHAMMET NURULLAH ÇETER

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
Danışman: Yrd. Doç. Dr. Sevcan YILMAZ GÜNDÜZ

Eskişehir
Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
ARALIK, 2017

JÜRİ VE ENSTİTÜ ONAYI

Muhammet Nurullah Çeter'in "***Ağ Saldırı Tespitinde Sınıflandırma Algoritmalarının Karşılaştırılması***" başlıklı tezi 22/12/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca, **Bilgisayar Mühendisliği** Anabilim dalında Yüksek Lisans Yeterlik tezi olarak kabul edilmiştir.

	Unvanı-Adı Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yrd. Doç. Dr. Sevcan YILMAZ GÜNDÜZ**	**…….....**
Üye	**: Yrd. Doç. Dr. Ahmet ARSLAN**	**……..**
Üye	**: Yrd. Doç. Dr.Mehmet KOÇ**	**……..**

……..

Enstitü Müdürü

ÖZET

AĞ SALDIRI TESPİTİNDE SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

Muhammet Nurullah ÇETER
Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Aralık, 2017
Danışman: Yrd. Doç. Dr. Sevcan YILMAZ GÜNDÜZ

Günümüzde bilişim sistemlerinde ağ güvenliği çok önemli bir duruma gelmiştir. İnsanlar ve kurumlar aralarındaki iletişimin büyük bir kısmını bilgisayar ağları üzerinde gerçekleştirmektedir. Bu ağdaki iletişim esnasında gizli bilgilerimiz de bulunabilir. Gizlilik, bütünlük, erişilebilirlik bilgilerimiz için çok önemlidir. Kötü niyetli kişiler ağ üzerinde bulunan güvenlik açıklarından faydalanarak bilgilerimizi çalabilir ya da bilişim sistemlerimizi kullanamaz hala getirebilir. Bilişim sistemlerinde ağ üzerinde yapılan bu tür saldırılara karşı korunmak için günümüzde saldırı tespit sistemleri geliştirilmiştir. Bu noktada saldırı tespit sistemlerinde kullanılan algoritmalar büyük bir önem teşkil etmektedir. Çünkü bu algoritmalar performans açısından farklılıklar göstermektedir. Bu araştırmada 4 farklı makine öğrenme algoritması Waikato Environment for Knowledge Analysis (WEKA) ortamında kullanıldı. Bu öğrenme algoritmaları çok katmanlı sinir ağları(MLP), destek vektör makineleri(SVM), karar ağacı (J48), bulanık düzensiz kural indüksiyon(FURIA) algoritmalarıdır. Bu tezde saldırı tespit sisteminde verilen algoritmalar performans açısından karşılaştırıldı.

Anahtar Sözcükler: Ağ Saldırısı, Öğrenme Algoritmaları, WEKA, Saldırı Tespit Sistemleri, Ağ Güvenliği

ABSTRACT

COMPARISION OF CLASSIFICATION ALGORITHMS FOR NETWORK INTRUSION DETECTION

Muhammet Nurullah ÇETER

Department of Computer Engineering
Anadolu University, Graduate School of Science, December, 2017
Supervisor:Asst.Prof.Dr. Sevcan YILMAZ GÜNDÜZ

Network security has become very important nowadays in information system. A large part of the communication between people and institutions is realized on computer networks.We may also have confidential information during this network communication. Confidentiality, integrity, availability is crucial to our knowledge. Malicious people can still our information or take advantage of our information systems by exploiting vulnerabilites on the network. Intrusion detection systems have been developed to protect against such attacks on networked systems today's. At this point algorithms used in intrusion detection systems are of great importance. Because these algorithms vary in terms of performance. In this research, 4 different machine learning algorithms were discovered in WEKA. These learning algorithms are multi layer perceptrons (MLP), support vector machines (SVM), decision tree algorithm (J48) and fuzzy unordered rule induction algorithm (FURIA). In this thesis, these algorithms were compared against each other in terms of performance in the intrusion detection system.

Keywords: Network security, Machine Learning algorithm, WEKA, Intrusion Detection System, Unauthorized Access

TEŞEKKÜR

Çalışmamı yaptığım süre boyunca bilgi ve deneyimleriyle bana yol gösteren, değerli tez danışmanım sayın Yard. Doç. Dr. Sevcan Yılmaz Gündüz'e en içten teşekkürlerimi sunarım.

Aynı zamanda tez çalışmalarım süresince desteklerini esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

Muhammet Nurullah ÇETER

Aralık 2017

…../…../20

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalardan bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan "bilimsel intihal tespit programıyla tarandığını ve hiçbir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

…….

.. …….

(Öğrencinin Adı Soyadı)

İÇİNDEKİLER

Sayfa

TABLOLAR DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

BFS : Best First Search (En İyi İlk Arama)

CFS : Correlation-based Feature Selection
(Korelasyona Dayalı Özellik Seçimi)

DARPA : Defense Advanced Research Projects Agency
(Savunma İleri Araştırma Projeleri Ajansı)

FURIA : Fuzzy Unordered Rule Induction Algorithm
(Bulanık Düzensiz Kural İndüksiyon Algoritması)

IP : Internet Protocol(İnternet Protokolü)

IPV4 : Internet Protocol Version 4 (İnternet Protokolü version 4)

IPV6 : Internet Protocol Version 6 (İnternet Protokolü version 6)

KDD CUP'99 : Knowlodge Discovery and Data Mining Tools Competition 1999
(Bilgi Buluşması ve Veri Madenciliği Araçları Yarışması 1999)

MAE : Mean Absolute Error(Ortalama Mutlak Hata)

MLP : Multi Layer Perceptron (Çok Katmanlı Sinir Ağları)

RMSE : Root Mean Square Error(Kök Ortalama Kare Hata)

STS : Saldırı Tespit Sistemi

SVM : Support Vector Machines(Destek Vektör Makineleri)

TCP : Transmission Control Protocol(Taşıma Kontrol Protokolü)

UDP : User Datagram Protocol(Kullanıcı Veri Protokolü)

WEKA : Waikato Environment for Knowledge Analysis
(Bilgi Analizi için Waikato Ortamı)

1.GİRİŞ

Günümüzde internetin yaygınlaşması ile birlikte iletişim ağları üzerindeki saldırılar da artmaktadır. Bu saldırılar değişik şekillerde yapılmaktadır. İnternette gezinen saldırganlar sistemlerin değişik açıklarını bularak bu yöntemlerle sistemlere saldırmaya başlarlar. Saldırılar sonucu kurum ya da kişisel bilgisayarlarda bulunan bilgiler çalınmakta ya da internet üzerinden verilen hizmetlerde büyük sorunlara neden olmaktadır. Kurumlara ve kişilere yapılan bu siber saldırılar kurum ve kişilerin itibarını olumsuz bir şekilde etkilemektedir. Bilişim ağlarındaki bu saldırıların önlemesi için saldırı tespit sistemleri geliştirilmiştir. Saldırı tespit sistemleri bütün ağ trafiğini denetler. Gelen ve giden bağlantılarda şüpheli durumları tanımlar. Saldırı tespit sistemlerinin asıl fonksiyonu bilgisayar ya da bilgisayar ağlarının gizliliği, bütünlüğü ve erişilebilirliğini tehlikeye sokan yetkisiz girişimleri tespit etmektir (Alamleh,2015). Saldırı tespit sistemlerinin oluşturulmasında istatistik, yapay zeka, veri madenciliği gibi farklı yöntemler kullanılmıştır. Saldırı tespit sistemleri saldırıyı tespit etme yöntemine göre imza tabanlı saldırı tespit bir de anormallik tabanlı saldırı tespit olarak iki temel sisteme ayrılır. İmza tabanlı saldırı tespit sistemleri genelde bilinen saldırıları tespit ederken, anomali tabanlı saldırı tespit sistemleri bilinmeyen saldırıları da tespit etmede yardımcı olur. Saldırı tespit sistemleri aynı zamanda saldırıların sınıflandırılmasını sağlar.

Bu konu ile ilgili değişik çalışmalar yapılmıştır. Tanrıkulu çalışmasında DARPA veri setini kullanarak ağ üzerinde bilgi sistemlerine yapılan saldırıların tespitinde yapay sinir ağlarının kullanılması araştırmış, yüksek başarım tespit etmiş ve örnek bir saldırı tespit sistemi oluşturmuştur (Tanrıkulu, 2009). (Sahu, Mehtre,2015)' de Kyoto 2006+ veri seti ile J48 karar ağacı algoritması saldırı tespit etmede kullanılmış ve performans olarak %97.23 doğru bir sonuç alınmıştır (Mukherjeea, Sharma, 2012)'de Naïve Bayes sınıflandırma algoritması farklı özellik seçimleri kullanılarak en iyi sonucun korelasyona dayalı özellik seçimi (CFS) olduğu ortaya konulmuştur. (Belavagi, Muniyal, 2016)'de 4 farklı algoritma Lojistik Regrasyon, Destek Vektör Makineleri, Naif Bayes ve Rastgele Orman algoritmaları WEKA ortamında karşılaştırılmıştır. Bunlar arasında en iyi algoritma yüzde 99 doğruluk oranı ile Rastgele Orman algoritması olmuştur.

Bu tezde saldırı tespit sistemlerinde farklı makine öğrenme algoritmaları kullanılarak performans açısından farklı sonuçların oluştuğu gözlemlendi. Makine öğrenme algoritmalarından çok katmanlı sinir ağları(MLP), destek vektör

makineleri(SVM), karar ağacı(J48), bulanık düzensiz kural indüksiyon (FURIA) algoritmaları Waikato Environment for Knowledge Analysis(WEKA) ortamı kullanılarak performans açısından karşılaştırmalar yapıldı. Bu karşılaştırmalar yapılırken değerlendirme ölçütleri olarak Ortalama Mutlak Hata(MAE), Kök Ortalama Kare Hata(RMSE), ve karışıklık matrisi sonucu elde edilen doğru sınıflandırılan örneklerin yüzdesi temel alınmıştır. Ayrıca her bir algoritmanın çalışma süreleri hesaplanarak zaman açısından performans karşılaştırması yapılmıştır. Tezin ikinci kısmı olarak özellik seçme işlemi yapılmıştır. Hangi algoritma için hangi özellik kümesinin performansının daha iyi olduğu araştırılmıştır.

2.BİLGİSAYAR AĞLARI ve IP

Bütün bilişim sistemlerini birbirine bağlayan internet ağı sayesinde günümüzde insanlar çok uzak noktalardan birbiriyle iletişim sağlayabilmektedirler. İnternet üzerinde bilgilerin iletişim ve kontrolünü, iletişim kontrol protokolü(TCP) ve paketlerin adresleyerek iletmesini sağlayan internet protokolü(IP) ile sağlanmaktadır.

2.1. İnternet Protokolü

İnternet'te her bilgisayarın bir IP adresi bulunmaktadır. IP adresi sayesinde internet üzerinde bulunan tüm bilgisayarlar birbiriyle kolayca iletişime geçerler. IP adreslerini akılda tutmak zor olduğundan bunlara karşılık gelen bir isim ile tanımlanmıştır. Örneğin 10.45.0.200 rakamları antivirus.anadolu.edu.tr ile tanımlanmıştır. İnternete bağlı cihazlar arasındaki iletişim IP protokolü sayesinde olur. İnternet çok sayıda bilgisayarı birbirine birleştiren ağ topluluğudur. IP'ye diğer deyimiyle internetin ortak dili denilebilir. Ağ üzerindeki paketlerin adresleyerek iletişimi ve yönlendirilmesinden sorumludur. Bazı IP adresleri özel kullanımlar için ayrılmıştır.

2.2 IPv4

Günümüzde kullandığımız 4 tane sekizlik bitten oluşan standart internet protokolüdür. Bu sayılar, 0 ila 255 arasında değişmektedir. IPV4 A, B, C, D, E olmak üzere 5 sınıftan oluşur (Çölkesen ve Örencik, 2002). A sınıfı 1.x.x.x ile 126.x.x.x arasındaki rakamları kapsar.126 tane ağ vardır. Her ağdaki sunucu sayısı 16.777.224'tür. B sınıfı 128.0.x.x ile 191.255.x.x arasındaki sayıları kapsar. B sınıfında 16384 ağ bulunmaktadı

Her ağda 65534 tane sunucu bulunur. C sınıfı 192.0.0.x ile 223.255.255.x arasındaki sayıları kapsar. C sınıfı 2097152 tane ağ bulunur. Her ağda 254 tane sunucu bulunur D sınıfı 224.0.0.0 ile 239.255.255.255 arası kapsar. E sınıfı 240.0.0.0 ile 239.255.255.255 arası kapsar. Bu protokol kullanılarak 4 milyardan fazla adres üretilebilmektedir.

Şekil 2.1. *Ipv4 Başlık Yapısı*

2.3 IPv6

Günümüzde bilişimde kullanılan sistemlerin ve kullanıcı sayısının artmasıyla ipv4 adres sayısı her geçen gün azalmaktadır. Bu nedenle ipv4'in yerini alacak ipv6 kullanılmaya başlanmıştır.Ipv4 sayısı yaklaşık 4.3 milyar tanedir. Ipv6'yı IPv4'den ayıran en önemli kısmı adres uzunluğudur. Ipv6, 128 bit genişliğindedir. 2^{128} adet 3 x 10^{38} tane adres demektir. Ipv6 paket yapısı Ipv4'ten farklıdır (Çölkesen ve Örencik, 2002).

2.4 İletim Kontrol Protokolü

Ağ üzerindeki bir paketin bütünlüğünü ve iletişim garantisini sağlayan protokoldür. TCP'de tanımlı temel görevler üst katmandan gelen verinin uygun parçalara bölünmesi, her parçaya, alıcı kısmında aynı biçimde sıraya koyulabilmesi amacıyla sıra numarası verilmesi ,kaybolan ve bozuk gelen parçaların tekrarlanması olarak sıralanır (Çölkesen ve Örencik, 2002). Burada üçlü el sıkışma denilen mantık çalışır. İstemci önce sunucuya istekte bulunur. Sunucu istemciye isteği aldığına dair bir uyarı gönderir. En son istemci tekrar sunucuya bir mesaj iletir. Böylelikle güvenli iletişim başlamış olur. Dosya transfer

protokolü(FTP), Elektronik posta gönderme protokolü(SMTP), hiper metin transfer protokolü(HTTP), güvenli hiper metin transfer protokolü(HTTPS), güvenli kabuk(SSH), e-posta protokolü(POP3) gibi birçok protokollerin veri iletimi TCP aracılığıyla olur.

2.5. Kullanıcı Veri Protokolü

Bu protokol de TCP protokolü gibi taşıma katmanı protokollerinden birisidir. Veri iletimi sağlar. TCP protokolüne göre daha az güvenlidir. Ağ üzerinde paketi gönderir ama paketin yerine ulaşıp ulaşmadığını kontrol etmez. Alan adı sistemi(DNS), küçük dosya iletim protokolü(TFTP),basit ağ yönetim protokolü (SNMP) gibi bazı protokoller buna örnek verilebilir. Bu protokol TCP protokolüne göre daha hızlıdır. Bunun için ses ve video gönderiminde bu protokol kullanılır.

3.SALDIRI TESPİT SİSTEMLERİ(STS)

Saldırı bilginin gizliliğini, bütünlüğünü ve erişilebilirliğinin tehdit edilmesi ve bozulması yönünde yapılan her türlü girişime denilebilir. Günümüzde bilişim sistemlerine çok değişik saldırılar yapılmaktadır. Bu saldırıları önlemek için saldırı tespit sistemleri geliştirilmiştir. Saldırı tespit sistemleri, ağ üzerindeki bilginin gizliliğini, bütünlüğünü, ve erişilebilirliğinin bozulması ve tehdit altında kalması riskine karşı üçüncü şahıs ve ya sistemler tarafından izlenerek buna tedbir alınmasını sağlayan sistemlerdir.

3.1. Saldırı Tespit Sistemlerinin Sınıflandırılması

Saldırıları tespit ettikleri ortama ve tespit yöntemlerine göre ikiye ayrılır.

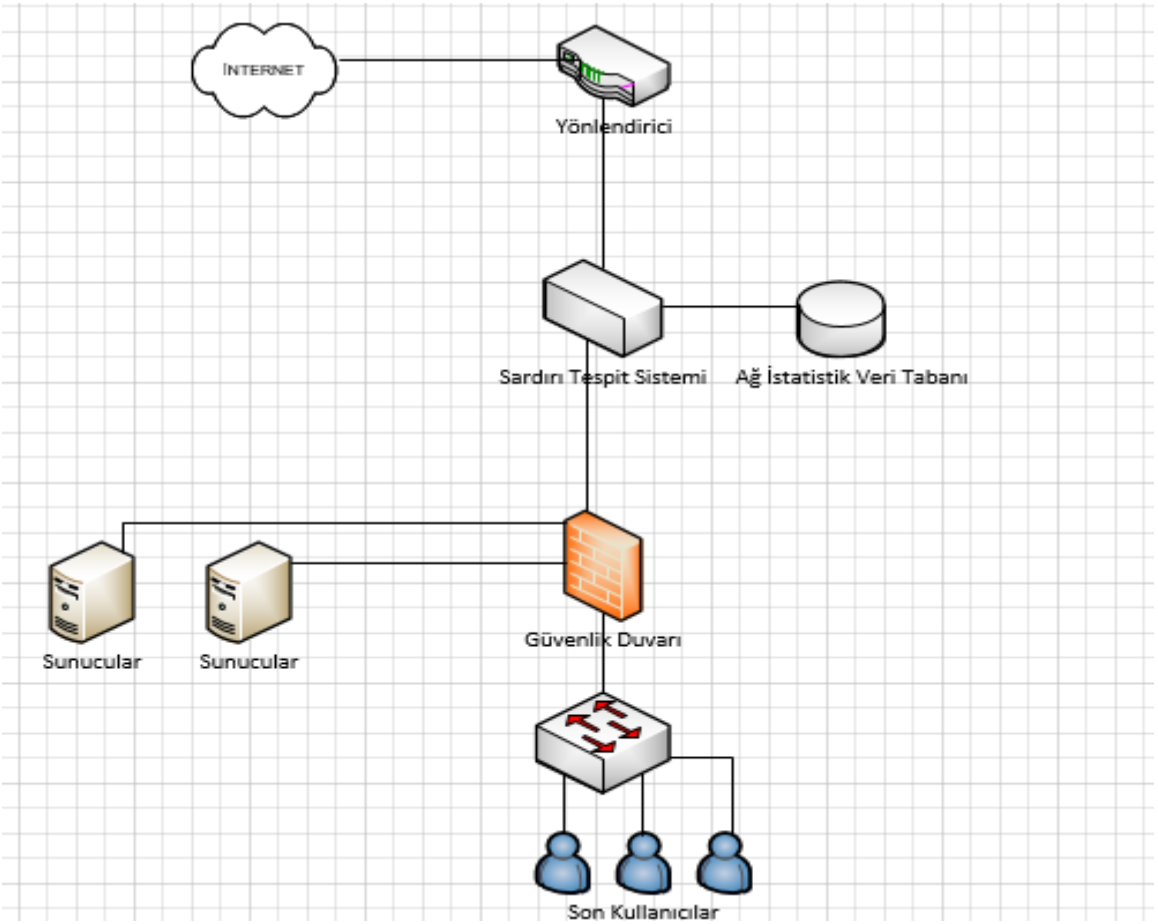
3.1.1.Ortama göre STS

Saldırı tespit sistemi ortama göre sunucu tabanlı ve ağ tabanlı olmak üzere ikiye ayrılır. Sunucu tabanlı saldırı tespit sistemi sadece o sunucuda ya da bilgisayarda saldırıları tespit eder. Ağ tabanlı saldırı tespit sistemi ise o ağda bulunan bütün saldırıların tespit edilmesinde rol oynar.

3.1.2.Tespit yöntemine göre STS

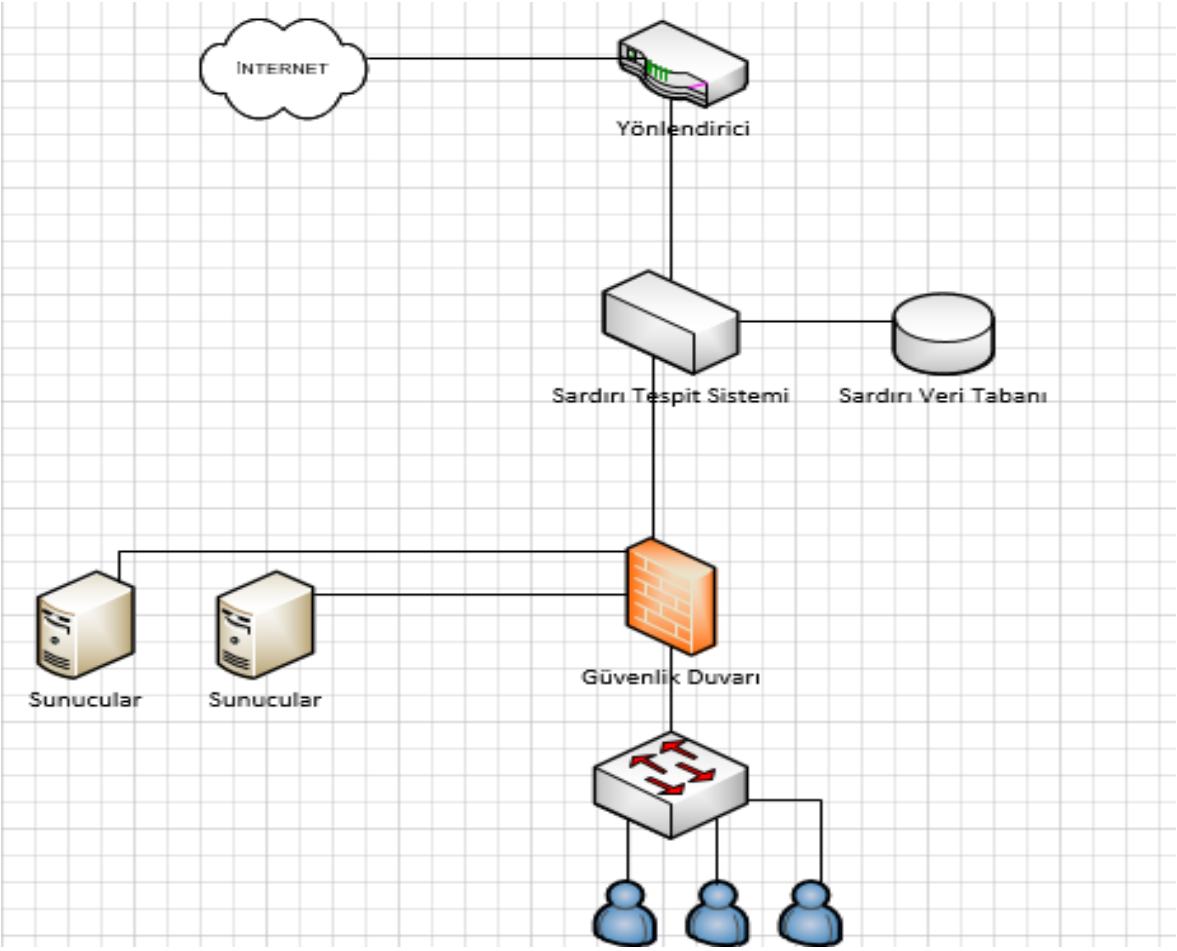
Saldırı Tespit sistemleri saldırıyı tespit yöntemine göre iki ana sınıfa ayrılmaktadır. Bunlar İmza tabanlı saldırı tespit sistemi ve anomali tabanlı saldırı tespit sistemi olarak söyleyebiliriz. İmza tabanlı saldırı tespit sitemi veri tabanında yer alan imzalarla karşılaştırılarak bunun bir saldırı olup olmadığına karar verir. İmza tabanlı saldırı tespit

sistemleri var olan saldırıları tespit eder yani yeni olan bir saldırıyı veri tabanında olmadığı için tespit edemez(Manzoor, Kumar, 2017). Anomali tabanlı saldırı tespit sistemleri ağ trafiğindeki normal olmayan durumları tespit ederek bunun bir saldırı olabileceğini tespit eder. Anomali tabanlı saldırı tespit sistemleri yeni saldırıları tespit edebilir (Agrawal, Agrawal, 2015). Anomali tabanlı saldırı tespit sistemi herhangi bir saldırı imzası kullanmadığı için yanlış uyarı verme ihtimali daha yüksektir. Anomali tabanlı saldırı tespit sistemi Şekil 3.1'de imza tabanlı saldırı tespit sistemi de Şekil 3.2'de gösterilmiştir.



Şekil 3.1. *Anomali Tabanlı Saldırı Tespit Sistemi*

Şekil 3.1'de görüldüğü üzere sunucular ve kullanıcılar internete çıkarken bunların ağ istatistikleri veri tabanında saklanıyor. Genel olarak bu ağ istatistikleri birbirine yakın değerler gözlemlenmektedir. Ancak ağ istatistik veri tabanında ağ trafiğinde görülen ani artışlar saldırı tespit sistemi bunun bir saldırı olduğuna işaret eder (Alamleh,2015).



Şekil 3.2. *İmza Tabanlı Saldırı Tespit Sistemi*

Şekil 3.2'de görüldüğü üzere son kullanıcılar ve sunucular internete çıkarken veya internetten bunlara gelen isteklerde saldırı tespit sisteminde bu kullanıcılara ve sunuculara ait bilgileri gözlemleyerek saldırı veri tabanında bulunan imzalarla ile karşılaştırma yaparak bunların bir saldırıya maruz olup olmadığını tespit eder (Alamleh,2015).

4. AĞ SALDIRILARINI SINIFLAMADA KULLANILAN ALGORİTMALAR

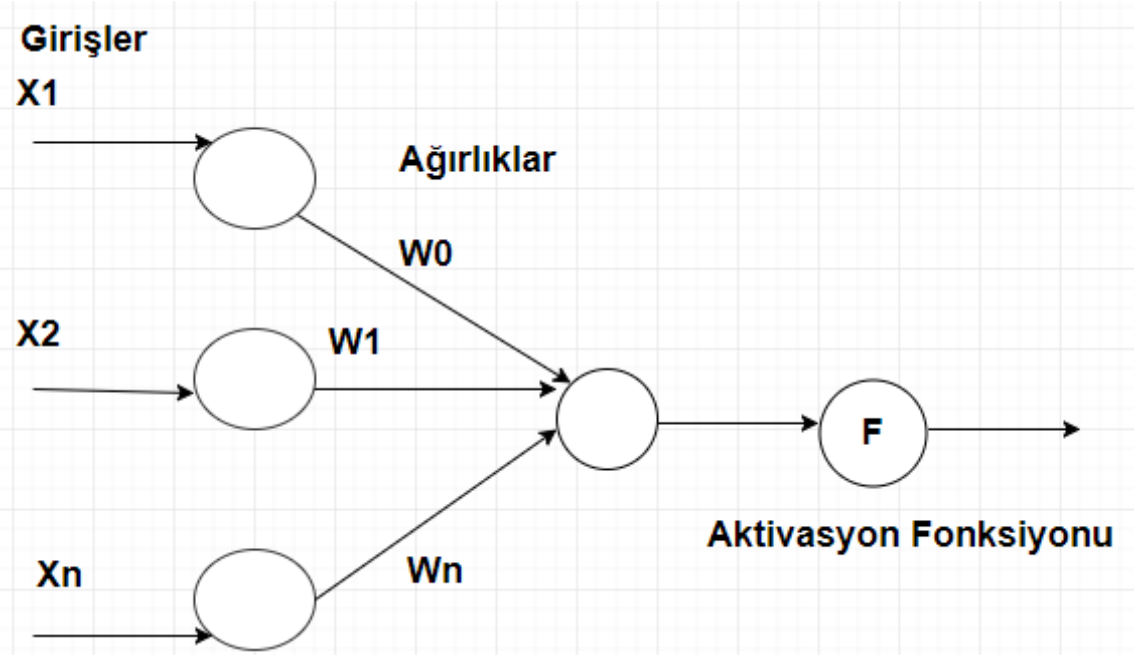
4.1. Yapay Sinir Ağları

Ağ saldırı tespit sistemlerinde farklı algoritmalar kullanılmaktadır. Yapay sinir ağları insan beyninden esinlenerek geliştirilmiştir. Önemli makine öğrenme algoritmalarından biridir. Bu öğrenme algoritması tek katmanlı veya çok katmanlı olabilir. Bu ağlar öğrenme, hafızaya alma ve veriler arasında ilişki çıkarmasını sağlar.

Yapay sinir ağları giriş katmanı, gizli katman ve çıktı katmanı olmak üzere üç ana katmandan oluşur.

4.1.1.Tek katmanlı sinir ağları

Bir sinir hücresinin birden fazla girdi alarak bir çıktı üretmesini sağlar. Yapay sinir ağlarında en basit tek katmanlı sinir ağı modelidir. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal bir fonksiyondur. Fonksiyon çıktısı +1 ve ya -1 değerini alır. Tek katmanlı sinir ağlarını ifade eden durum Şekil 4.1'de gösterilmiştir.



Şekil 4.1. *Tek Katmanlı Sinir Ağı*

Şekil 4.1'de X1, X2, Xn giriş değerlerini w0, w1, wn değerleri ağırlıkları F aktivasyon fonksiyonunu ifade etmektedir.

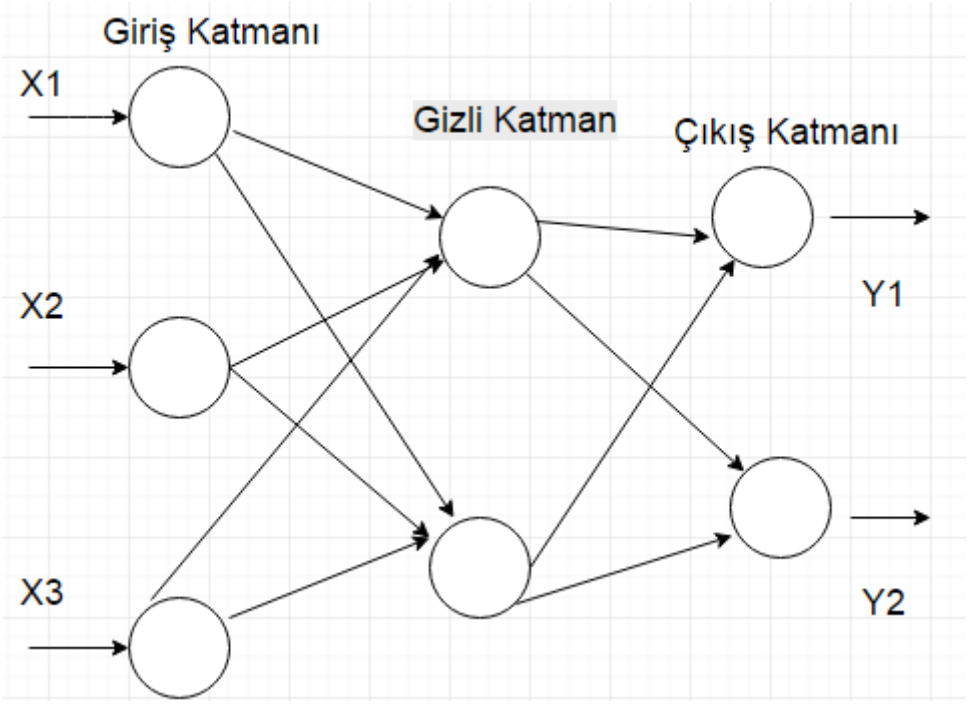
$$y_i = f(\sum_{k=1}^{n} w_k x_k + w_0) \tag{4.1}$$

4.1 denkleminde belirtilen y_i çıkışları, w_k ağırlıkları, x_k girişleri, w_0 ilk ağırlık değerini ifade etmektedir.

4.1.2. MLP algoritması

Çok katmanlı ağlar ara katmanın olması nedeniyle tek katmanlı sinir ağlarından ayrılır. Çok katmanlı sinir ağları en az üç ve ya daha fazla katmandan oluşur (Singh,

Bansal,2013). Ara katman gizli katman olarak da bilinir. Girdi katmanından gelen veriler burada işlenir. Çok katmanlı sinir ağlarını ifade eden durum Şekil 4.2'de ifade edilmiştir.



Şekil 4.2. *Çok Katmanlı Sinir Ağı*

Şekil 4.2'de X1, X2, X3 giriş değerlerini, Y1 ve Y2 çıkışları ifade etmektedir.Çok katmanlı sinir ağları algoritmalarında değişik aktivasyon fonksiyonları kullanılabilir. Bunlar sigmoid ve tanh(x) fonksiyonları olabilir. Denklem 4.2 ve Denklem 4.3'te sırasıyla sigmoid ve hiperbolik tanjant fonksiyonu verilmiştir.

$$y = \frac{1}{1+e^{-net}} \tag{4.2}$$

$$y = \frac{1-e^{-2net}}{1+e^{2net}} \tag{4.3}$$

Çok katmanlı sinir ağlarında ileri beslemeli ve geriye yayılım algoritması kullanılır (A. Aziz , Hanafi , Hassanien,2017). Çok katmanlı ağın çıkışı aşağıda verilen denklem ile hesaplanır.

$$y_k = F_k[\sum_{m=1}^{n_h} b_{k,m} f_m(\sum_{l=1}^{n_x} w_{m,l} x_l + w_{m,0}) + b_{k,0}] \tag{4.4}$$

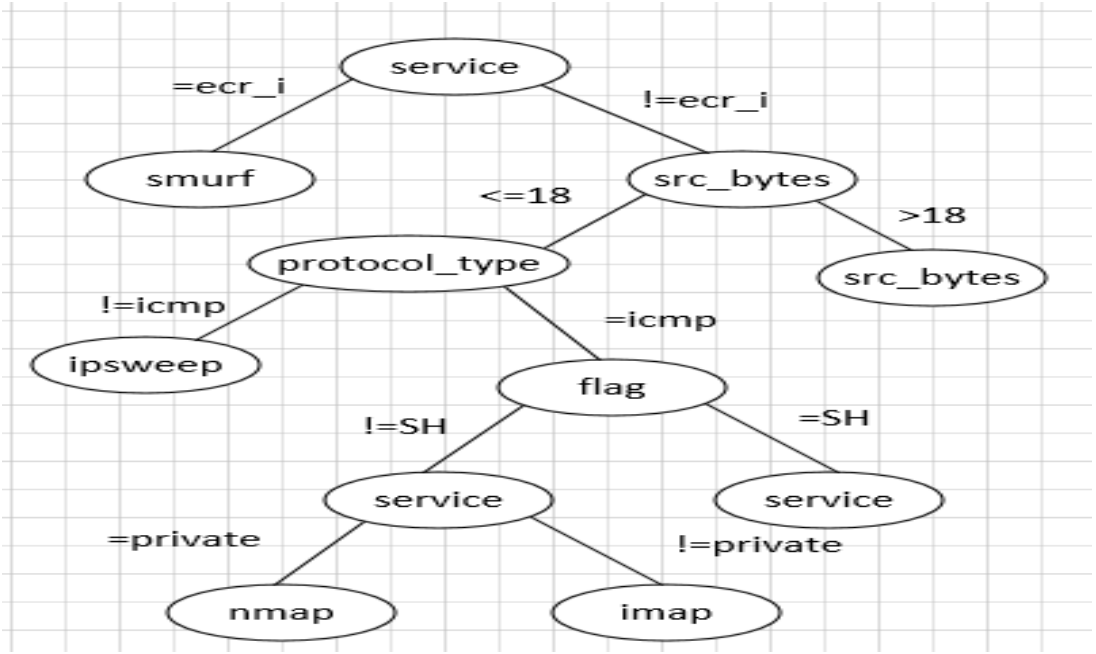
Denklem 4.4'de belirtilen y çıkışları, b sapmayı, w ağırlıkları, x girişleri ifade ediyor.

$$\Delta w_{ij}(t+1) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(t) \tag{4.5}$$

Denklem 4.5'te belirtilen geri yayılım algoritmasında ağırlık güncellemesinde kullanılan w_{ij} ağırlıkları, η öğrenme katsayısını, $\frac{\partial E}{\partial w_{ij}}$ ifadesi gradient,

4.2. J48 Karar Ağacı Algoritması

Karar ağacı öğrenmesi, sınıflandırmada kullanılan önemli makine öğrenme algoritmalarından bir tanesidir. Karar ağaçları öğrenme ve veri madenciliğinde en çok kullanılan yöntemlerinden birisidir. Karar ağaçlarının çok kullanılmasının nedenleri arasında eğitimi ve test edilmesinin hızlı olması, yorumlanmasının kolay olması, görselliğin iyi olmasıdır(Özgür, Erdem,2012). Bu tezde karar ağaçları algoritmalarından J48 algoritması kullanıldı. Bu algoritmanın diğer adı C4.5 olarak geçmektedir.J48 algoritmasında her adımda bütün özellikler kontrol edilir. Sonra her özelliğin bilgi kazanımı(information gain) hesaplanır. En iyi bilgi kazanımı veren özellik karar ağacında karar olarak belirlenir. Bilgi kazancı ölçümünde Entropy kullanılır (Meena, Choudhary, 2017). Entropy rastgeleleri, belirsizliği ve beklenmeyen durumun ortaya çıkma ihtimalini gösterir. Ardından oluşan karar düğümün altında bir alt liste oluşturularak alt karar ağacı inşa edilir.



Şekil 4.3. *J48 Algoritması ağaç yapısı örneği*

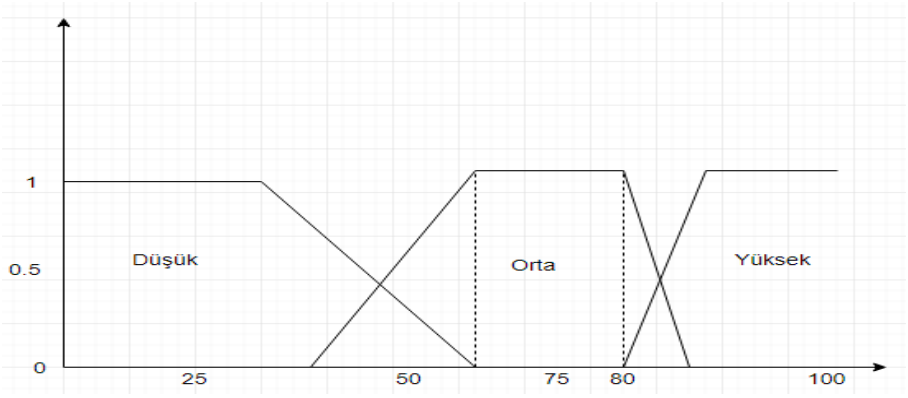
Şekil 4.3 J48 algoritmasında kullanılan ağaç yapısı bir örnek üzerinde ifade edilmiştir. Burada service özelliği entropi değeri en yüksek olduğu için en başa yazılmış ve karar ağacının çizimine devam edilmiş.

$$H(x) = E(I(X)) = \sum_{i=1}^{n} p(x_i)\log_2(1/\,p(x_i)) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \qquad (4.6)$$

Yukarıdaki belirtilen 4.6'daki entropi denkleminde $p(x_i)$ herhangi bir özelliğin frekansını gösteren değerdir. Entropi değeri en yüksek çıkan özellik karar ağacında en üste yazılır ve bu şekilde karar ağacının çizimine devam edilir (Lin, Ying , Lee, 2012).

4.3. Bulanık Mantık Tabanlı Algoritma FURIA

Klasik mantıkta sıcak, soğuk, hızlı yavaş gibi kavramlar vardır. Fakat sıcak ile soğuk arasındaki ılık kavramı, hızlı ile yavaş arasındaki orta hızlı kavramı ya da öğrencilerin başarı durumuyla ilgili düşük, yüksek kavramlarının dışında orta gibi kavramlar denilince bulanık mantık devreye girer. Bulanık mantıkta kümeye ait her bir eleman [0 1] arasında üyelik dereceleri alır. Öğrenci başarı durumundaki bulanık mantığı ifade eden durum Şekil 4.4 ile gösterilmiştir.



Şekil 4.4. *Öğrenci Başarı Durumunu Gösteren Bulanık Küme Gösterimi*

Bu tezde bulanık mantık tabanlı FURIA algoritması sınıflandırma için kullanılmıştır. FURIA algoritması RIPPER algoritmasını temel almıştır(Barthakur, Dahal, Ghose,2015). Bu algoritma RIPPER algoritmasının değiştirilmiş ve geliştirilmiş halidir. Bu algoritma Jens Christian Huehn, Eyke Huellermeier kişileri tarafından 2009 yılından oluşturulmuştur (Hühn, Hüllermeier, 2005).

$$I^F = (\phi^{s,L}, \phi^{c,L}, \phi^{c,U}, \phi^{s,U}):$$

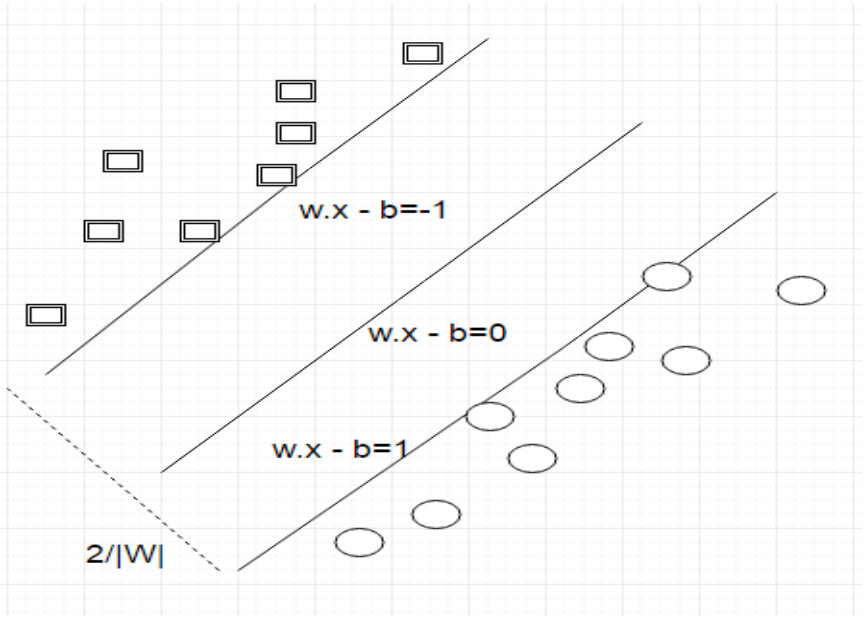
$$I^F(v) = \begin{cases} 1 & \phi^{c,L} \le v \le \phi^{c,U} \\ \dfrac{v - \phi^{s,L}}{\phi^{c,L} - \phi^{s,L}} & \phi^{s,L} < v < \phi^{c,L} \\ \dfrac{\phi^{s,U} - v}{\phi^{s,U} - \phi^{c,U}} & \phi^{c,U} < v < \phi^{s,U} \\ 0 & \text{else} \end{cases} \tag{4.7}$$

Bulanık aralık 4.7 denkleminde, 4 tane parametre ile tanımlanmıştır. $\phi^{c,L}$ ve $\phi^{c,U}$ bulanık kümenin alt ve üst çekirdek sınırı oluşturur. Ayrıca $\phi^{s,L}$ ve $\phi^{s,U}$ sembolleri de alt ve üst sınırlarını desteklediği kısmı gösterir(Hühn, Hüllermeier, 2005). (src_bytes in [470, 13748, inf, inf]) and (service = http) => class=back. (CF = 0.99)

Yukarıdaki belirtilen ifade WEKA dan alınmış bir kuraldır. src_bytes ve service ifadeleri veri kümemizde bulunan özelliklerdir. Eğer bu özelliklerden src_bytes değeri 470 ile 13748 arasında ise ve service http ise FURIA algoritması bunu bir back saldırı tipi olarak tanımlamıştır.

4.4. Destek Vektör Makinaları Algoritması

Sınıflandırmada en çok kullanılan makine öğrenme algoritmalarından bir tanesidir. Sınıflandırmada birbirinden farklı iki grup arasına bir sınır çizerek iki gruba ayırmak mümkündür. Bu sınır iki grubun üyelerine en uzak olan yer olmalıdır. Destek vektör makinaları algoritması bu sınırı seçme konusunda bize yardımcı olur. Destek vektör makinalarında değişik çekirdek fonksiyonlar kullanılabilir. Bunlar doğrusal, polinom, Gauss, sigmoid olarak geçebilir (Horng, Su, Chen, Kao, Chen, Lai, Perkasa, 2011). Bu çekirdek fonksiyonlarından Gauss (RBF)fonksiyonu kullanıldı. Veriler doğrusal olarak ayrılamadığından veriyi doğrusal olmayan haritalama yaparak orijinal girdi uzayından daha yüksek boyuttaki bir uzaya aktarır. Doğrusal destek vektör makinelerin sınıflandırma mantığını ifade eden durum Şekil 4.5 ile gösterilmiştir.



Şekil 4.5. *Doğrusal destek vektör makinesi gösterimi*

Şekil 4.5'te belirtilen $w*x-b$ ayırıcı düzlem $b/\|w\|$ iki grup arasındaki mesafe farkını verir. Bu mesafe farkına marjin denilir. Marjin ne kadar büyük olursa sınıflandırma o

kadar iyi olur. Bu mesafe farkını en yüksek değere ulaştırmak için 0,-1,+1 değerlerine sahip üç doğruyu veren denklemde $2/\|w\|$ formülü kullanılmıştır. Doğrular arası mesafe 2 birim olarak belirlenmiştir (Raman, Somu, Kirthivasan, Liscano, Sriram, 2017).

$w*x-b=1$, $w*x-b=-1$, w:çoklu düzlemin ağırlık vektörü, b:sapma $(x_{i,}y_i)$ i=1…l , $x_i \in R^p$, $y_i \in$ {-1,+1} x vektör uzayımızda bir nokta y ise bunun -1 ve ya +1 olduğunu gösteren bir değerdir. Bu formülde γ,r ve d değerleri fonksiyon çekirdek parametresidir. $K(x_{i,}y_i)$ eğitim kümesini ifade ediyor. x_i ve x_j ifade edilen giriş değerleridir.

$$K(x_{i,}y_i)=\exp(-\gamma\|x_i-x_j\|^2),\gamma>0 \tag{4.7}$$

4.7 denkleminde belirtilen Radyal tabanlı fonksiyondur.

$$K(x_{i,}y_i)=x_i^T.x_j \tag{4.8}$$

4.8 denleminde belirtilen doğrusal fonksiyondur

$$K(x_{i,}y_i)=(\gamma x_i^T.x_j+r)^d \tag{4.9}$$

4.9 denkleminde belirtilen polinomiyal fonksiyondur.

$$K(x_{i,}y_i)=\tanh(\gamma x_i^T.x_j+r) \tag{4.10}$$

4.10 denleminde belirtilen sigmoid fonksiyondur

5.KDD CUP'99 VERİ SETİ

Saldırı tespit sistemlerinde en çok kullanılan veri setlerinden bir tanesidir. Bilgi bulma ve veri madenciliği konusunda beşinci uluslararası konferansla birlikte düzenlenen üçüncü uluslararası bilgi buluşması ve veri madenciliği araçları yarışması için kullanılan bir veri kümesidir. Aynı zamanda bu veri seti saldırı tespit sistemlerinde kullanılan en büyük veri seti olarak da bilinir. Bu veri setinde 5 milyona yakın kayıt bulunmaktadır. (Modi, Jain, 2015). Bu veri seti internet ortamından alındı.(http-1) Bu veri setinin oluşturulma süreciyle ilgili adımlar internet adresi üzerinde belirtilmiştir.(https-2) 4.899.500 veri seti içinde sayı olarak 5000 örnek seçimi rastgele yapıldı. Farklı saldırı tiplerine değinmek için de ana verinin farklı yerlerinden farklı sayıda saldırı tipi alındı. 4.899.500 veri seti içinde 22 çeşit saldırı bir de normal bağlantı olmak üzere 23 çeşit sınıf mevcuttur. Kullanılan deney verisi içinde 12 tane saldırı 1 tane normal olmak üzere 13 değişik sınıf mevcuttur.Her bir kaydın 41 tane özelliği bulunmaktadır. Bu özelliklerin

bazıları sayısal bazıları da metinsel ifadelerdir. Bu özellikler temel özellikler, içerik özellikleri, sunucu tabanlı trafik özellikleri, zamana bağlı trafik özellikleri olmak üzere 4 ana sınıfa ayırabiliriz (Bamakan, Wang, Yingjie, Shi, 2016). Zamana bağlı trafik özellikleri "aynı sunucu" ve "aynı servis" özellikleri kullanılarak çıkarılan özelliklere verilen isimdir. Aynı sunucu" özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantıların "aynı servis" özellikleri ise son iki saniye içerisinde aynı servise yapılan bağlantıların takip edilmesiyle elde edilir. Bununla birlikte, ana makineyi (veya portları) 2 saniyeden daha büyük bir zaman aralığı kullanarak tarayan birkaç yavaş tarama saldırısı var. Bu noktada mevcut bağlantı üzerinden durum ile ilgili olarak yüzdelik oranına bakılır. Bu veri setinde bulunan 22 çeşit saldırı tipi 4 ana sınıfa ayrılmıştır. Bu veri setinde bulunan özellikleri anlatan açıklamalar Tablo 5.1 ile ifade edilmiştir.

Tablo 5.1. *KDD99 veri setinde bulunan özelliklerin açıklaması (Lin, Ying , Lee, 2012).*

Özellik Adı	**Tanım**	**Tip**
duration	Bağlantının uzunluğu(sn)	Sürekli
protocol_type	Protokol tipi(tcp,udp)	Ayrık
service	Hedef servis tipi(telnet,ssh)	Ayrık
flag	Bayrak	Ayrık
src_bytes	Kaynaktan hedefe gönderilen veri	Sürekli
dst_bytes	Hedeften kaynağa gönderilen veri	Sürekli
land	Kaynak ve hedef IP aynı ise 1 değilse 0	Ayrık
wrong_fragment	Yanlış parçalama sayısı	Sürekli
urgent	Acil paket sayısı	Sürekli
hot	Sıcak göstergelerin sayısı	Sürekli
num_failed_logins	Başarısız olan girişlerin sayısı	Sürekli
logged_in	Eğer başarılı giriş yapılmışsa 1 değilse 0	Ayrık
num_compromised	Risk altında bulunan durumların sayısı	Sürekli
root_shell	root sheel elde edilmişse 1 değilse 0	Sürekli

[Tablo 5.1. *(Devam) KDD99 veri setinde bulunan özelliklerin açıklaması (Lin, Ying , Lee, 2012).*]**.**

su_attempted	"su root" komutu denenmişse 1 değilse 0	Sürekli
num_root	"root" olarak yapılan erişimlerin sayısı	Sürekli
num_file_creations	Dosya oluşturma işlemlerinin sayısı	Sürekli
num_shells	sheel olarak yapılan isteklerin sayısı	Sürekli
num_access_files	Erişim kontrol dosyalarındaki işlem sayısı	Sürekli
num_outbound_cmds	ftp oturumunda giden komutların sayısı	Sürekli
is_host_login	Eğer giriş sıcak listeye ait ise 1 değilse 0	Ayrık
is_guest_login	Eğer giriş misafir ise 1 değilse 0	Ayrık
count	Son iki saniyedeki mevcut bağlantı üzerinden aynı bilgisayara yapılan istek sayısı	Sürekli
srv_count	Son iki saniyedeki mevcut bağlantı üzerinde aynı servis üzerinden yapılan istek sayısı	Sürekli
serror_rate	Bağlantılarda SYN olarak alınan hataların yüzdesi	Sürekli
srv_serror_rate	Bağlantılarda aynı servis üzerinde SYN olarak alınan hataların yüzdesi	Sürekli
rerror_rate	Bağlantılarda aynı bilgisayar üzerinde REJ olarak alınan hataların yüzdesi	Sürekli
srv_rerror_rate	Bağlantılarda aynı servis üzerinde REJ olarak alınan hataların yüzdesi	Sürekli
same_srv_rate	Aynı servisteki bağlantıların yüzdesi	Sürekli
diff_srv_rate	Farklı servislerdeki bağlantıların yüzdesi	Sürekli
srv_diff_host_rate	Farklı bilgisayarlardaki aynı servis üzerindeki bağlantıların yüzdesi	Sürekli
dst_host_count	Aynı hedef bilgisayardaki bağlantıların sayısı	Sürekli

[**Tablo 5.1.** *(Devam) KDD99 veri setinde bulunan özelliklerin açıklaması (Lin, Ying , Lee, 2012).*].

dst_host_srv_count	Aynı hedef bilgisayara yapılan bağlantılardaki aynı servislerin sayısı	Sürekli
dst_host_same_srv_rate	Aynı hedef bilgisayara yapılan bağlantılardaki aynı servislerin sayısı	Sürekli
dst_host_diff_srv_rate	Mevcut bilgisayara yapılan farklı servislerin yüzdesi	Sürekli
dst_host_same_src_port_rate	Mevcut bilgisayarda aynı kaynakkapıdaki bağlantıların yüzdesi	Sürekli
dst_host_srv_diff_host_rate	Farklı bilgisayarlardan aynı servis üzerindeki bağlantıların yüzdesi	Sürekli
dst_host_serror_rate	Mevcut bilgisayardaki bağlantılarda S0 hatalarını yüzdesi	Sürekli
dst_host_srv_serror_rate	Mevcut bilgisayardaki bağlantılarda tanımlanan serviste S0 hatası alan bağlantıların yüzdesi	Sürekli
dst_host_rerror_rate	Mevcut bilgisayardaki bağlantılarda RST hatası alan bağlantıların yüzdesi	Sürekli
dst_host_srv_rerror_rate	Mevcut bilgisayardaki bağlantılarda tanımlanan serviste RST hatası alan bağlantıların yüzdesi	Sürekli
class	Bir durumun saldırı olup olmadığını saldırının adını ifaden eden simge	Ayrık

Tablo 5.1'de verilen özellikler ağ güvenliğinde temel alınan durumlardır. Bu özelliklere bakılarak ağda bir saldırı olup olmadığına karar verilir.

5.1. Saldırı Çeşitleri

Saldırıları dört ana sınıfa ayırıyoruz (Canedo, Maroño, Betanzos, 2011).

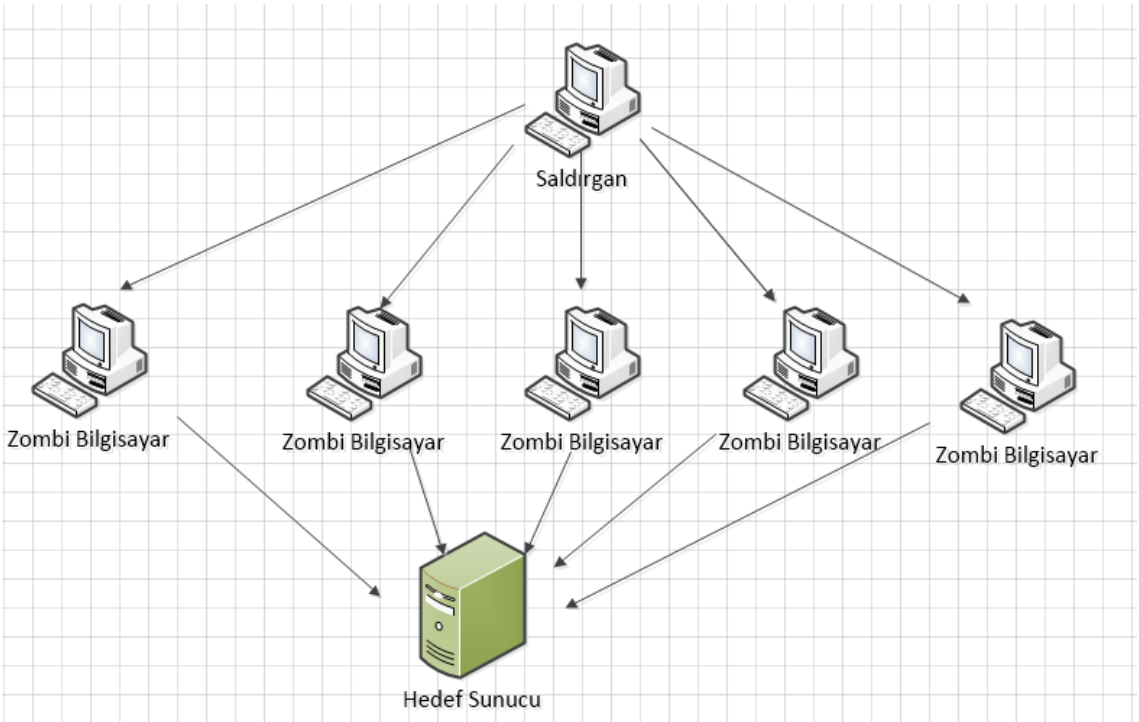
5.1.1. Bilgi tarama (Probe)

Bu saldırı tipinde bir sunucunun ya da herhangi bir bilgisayarın portlarını taranarak açık bulunan portlar bulunur. Böylece bu açık bulunan portlar sayesinde saldırgan rahatlıkla bu cihazlara saldırı yapabilir. Bu saldırılara örnek olarak ipsweep, belli bir portu

tarama saldırısı ya da portsweep, bir sunucu üzerinde verilen hizmetleri bulmak için sunucunun portlarını tarayan saldırılardır.

5.1.2. Hizmet dışı bırakma(Denial of Service-Dos)

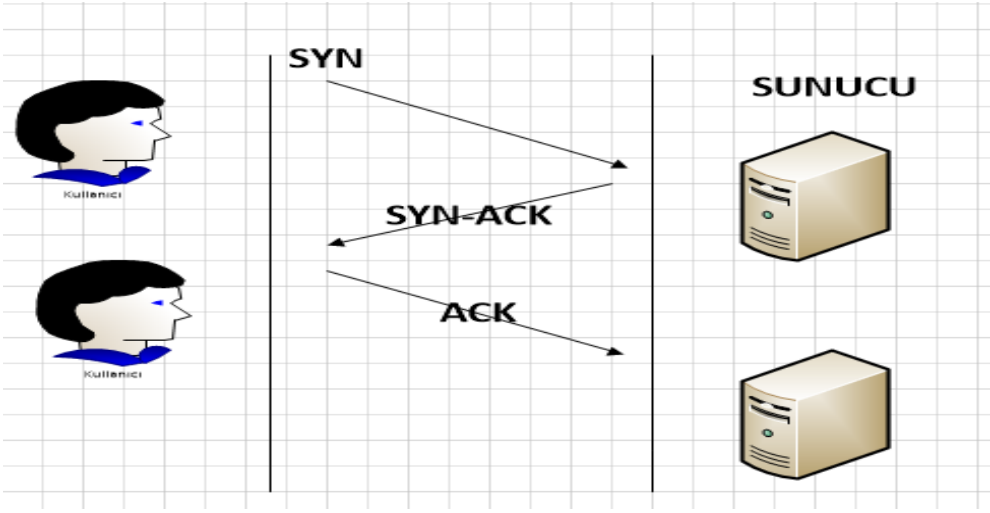
Bu saldırılar genelde sunucuya çok sayıda istek gönderilerek sunucunun dışarıya hizmet veremez hale getirilmesidir. Bu tür saldırılar tek bir makinadan yapılabileceği gibi denetim altına alınmış çok sayıda bilgisayar ile de yapılabilir. Denetim altına alınmış bu tür makinalara zombi bilgisayar diyoruz. Bu saldırılara smurf, selfping, tcpreset, mailbomb örnek verilebilir. Smurf, icmp paketlerinin broadcast ile ağ üzerindeki bütün cihazlara yayılmasıdır. Selfping saldırganın sürekli ping atmasıyla oluşur. Tcpreset saldırgan hedefteki sunucunun kurmaya çalıştığı bağlantıları için sunucu adına reset göndererek sunucunun bağlantısını engeller. Mailbomb saldırgan hedefteki sunucuya sürekli e-posta gönderir. Ddos saldırısını anlatan durum Şekil 5.1'de ifade edilmiştir.



Şekil 5.1. *Ddos saldırı örneği*

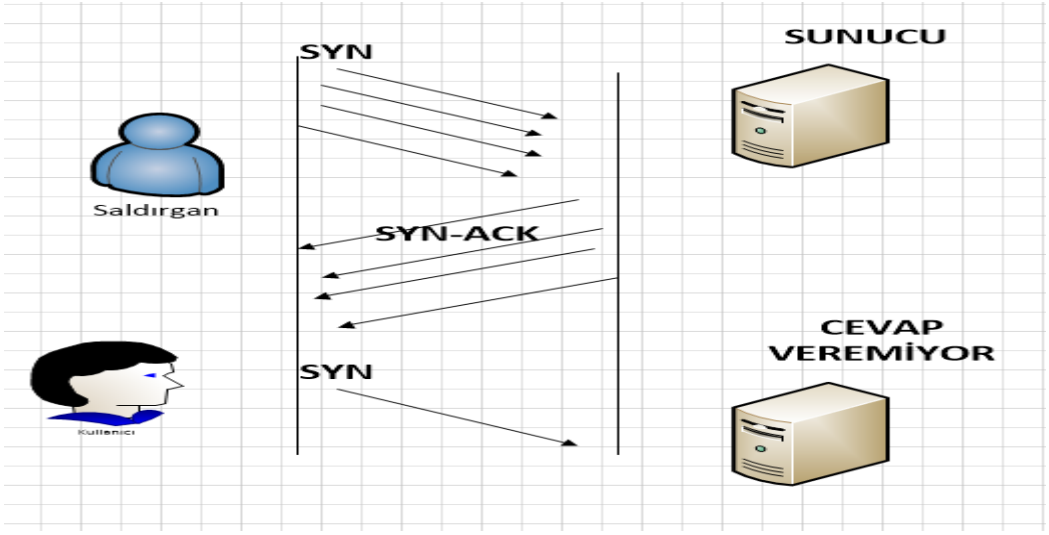
Şekil 5.1'de saldırgan kendisine seçtiği kurban bilgisayarları zombi bilgisayar haline getirmiştir. Yani daha önceden kurduğu bir zararlı yazılımla bu bilgisayarları denetimi altına almıştır. Sonra hedef sunucuya yapılan saldırıda toplu bir şekilde saldırarak hedef sunucunun hizmet yapamaz hale getirmesine neden olur. Bir iletişimin

sağlıklı bir şekilde olmasını anlatan üçlü el sıkışma modeli yapısı Şekil 5.2'de gösterilmiştir. SYN-ACK saldırısını anlatan durum Şekil 5.3'te gösterilmiştir.



Şekil 5.2. *Üçlü el sıkışma modeli yapısı*

Şekil 5.2'de belirtilen görselde bir kullanıcı ile sunucu arasındaki güvenli iletişim durumu ifade edilmiştir. Burada kullanıcı önce sunucuya SYN gönderir sonra sunucu kullanıcıya SYN-ACK en son da kullanıcı sunucuya ACK gönderir. Böylece üçlü el sıkışma dediğimiz durum oluşarak güvenli bir iletişim gerçekleşir.



Şekil 5.3. *SYN-ACK saldırı örneği*

Şekil 5.3'te belirtilen durum saldırgan sunucuya çok sayıda SYN paketi gönderir. Sunucu da saldırgana çok sayıda SYN-ACK paketi gönderir. Fakat saldırgan bu paketlere karşılık sunucuya herhangi bir cevap göndermez. Saldırganın yaptığı bu istekler için

sunucu çok sayıdan kaynak kullanır ve böylece sunucu gerçek kişilere hizmet veremez hale gelir.

5.1.3. Yönetici hesabı ile yerel oturum açma(Remote to Local-R2L)

Yönetici hesabı ile yerel oturum açma mevcut kullanıcı haklarının elde edilerek sisteme izinsiz erişim yapmaktır. Örnek olarak, ssh trojan saldırısı unix sistemlerde faaliyet gösteren bir saldırıdır. Guest basit şifreleri elde ederek sisteme girilmesi saldırısıdır.

5.1.4. Kullanıcı hesabının yönetici hesabına yükseltilmesi

Kullanıcı hesabının yönetici haklarını elde etmesi, yönetici haklarına sahip olmayan normal bir kullanıcının bazı açıklardan yararlanarak yönetici yetkilerini almasıdır. Bir örnek olarak sql saldırısı verilebilir. Sql saldırısı sql veri tabanı kurulu bilgisayarlarda sunucuya bağlanan saldırganın bazı komutlarla yönetici hakları elde etmesidir.

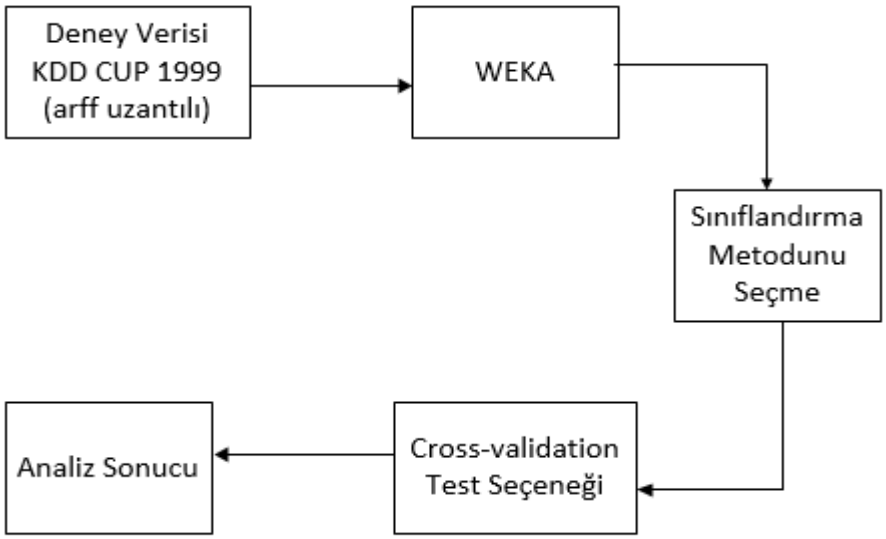
6.BİLGİ ANALİZİ İÇİN WAIKATO ORTAMI

WEKA çok sayıda makine öğrenme algoritması içeren java dili üzerinden geliştirilmiş açık kaynak kodlu bir yazılımdır(Kezih,Taibi,2013). Waikato üniversitesinde geliştirilmiştir. İsmini de buradan alır. WEKA üzerinde makine öğrenmesi ve istatistik ile ilgili çok sayıda hazır kütüphane bulunmaktadır. WEKA üzerinde veri ön işleme, sınıflandırma, gruplandırma, özellik seçimi ve özellik çıkarımını sağlayan seçenekler mevcuttur. WEKA ayrıca çıkan sonuçları görsel açıdan da gösteren bir yazılımdır. Bu çalışmada KDD99 veri setinden 5000 tane kayıt alınarak WEKA'da deneyler yapılmıştır. Bu deneylerde hangi öğrenme algoritmasının saldırıları daha iyi sınıflandırdığı ortaya konulmuştur. Bu ortamda veri seti comma separated value(csv) formatında ya da attribute-relation file format(arff) şeklinde çalışır(Aggarwala,Sharma, 2015). Bu çalışmada yapılan deneylerde arff formatı kullanılmıştır. WEKA'da verileri sınıflandırma yaparken 3 çeşit seçenek mevcuttur. Birincisi eğitim setini ve test setini ayrı yaparak önce eğitim setini eğitmek sonra test setini bu eğitim seti üzerinden dener. İkincisi veri kümesini istenilen sayıda kümeye böler. Alt kümelerden birini test diğerlerini eğitim kümesi kabul ederek sistemi eğitir. Sonra bu eğitim kümesini diğer bir alt küme üzerinden dener. Bu işlemi belirtilen küme sayısı kadar yaparak sistemi iyileştirmeye çalışır. Üçüncü seçenek veriyi yüzdelik dilime göre böler. İlk kümeyi eğitim diğer kümeyi test kümesi olarak kabul edip test yapar. Bu tezde, bütün deneyler ikinci seçenek olan

WEKA'da adı Cross-validation olarak geçen seçenek üzerinden yapıldı. Yani öncelikle veriyi 10 tane alt kümeye böler ve 9 parça öğrenme kümesi , 1 parça test kümesi kabul edilir(Alazab, Hobbs, Abawajy, Alazab,2012). Eğitim sonucunda oluşan model 1 parça üzerinde test edilir. Sonra test kümesi değiştirilerek diğer 9 parça eğitim kümesi olarak seçilir ve işlem bu şekilde devam eder. Yani 10-fold-cross-validation'da toplam 10 defa bu işlem yapılmış olur.

7. DENEYLER

Bu kısımda KDD99 veri setinden 5000 tane kayıt alındı. WEKA platformu kullanılarak sonuçlar tablolarda, grafiklerde gösterilerek karşılaştırmalar yapıldı. Tablolardaki karşılaştırmalarda doğru sınıflandırılan örnek sayıları, yüzdeleri, ortalama mutlak hata, kök ortalama kare hata değerlerine göre karşılaştırma yapılmıştır. Ayrıca Karışıklık Matrisi tablolarında hangi algoritmanın saldırıyı ne kadar sayıda doğru ve yanlış tahmin edildiği belirtilmiştir. Grafiklerde gerçek değerler ve öngörülen değerler olarak algoritmanın ne kadar iyi çalıştığı gözlemlendi. Sınıflandırma model yapısını anlatan durum Şekil 7.1 ile ifade edilmiştir. Deneylerde kullanılan saldırı tipi ve sayısı Tablo 7.1 ile ifade edilmiştir. Deneylerde kullanılan saldırı tipi ve kategorileri de Tablo 7.2 ile ifade edilmiştir.



Şekil 7.1. *Sınıflandırma Model Yapısı*

Tablo 7.1. *KDD CUP'99 veri setinde bulunan sınıf tipi ve sayısı*

Sınıf Tipi	**Sayısı**
normal	972781
back	2203
guess_passwd	53
imap	1081
ipsweep	12481
neptune	1072017
nmap	2316
portsweep	10413
rootkit	10
satan	15892
smurf	2807886
teardrop	979
warezclient	1020
land	21
warezmaster	20
loadmodule	9
ftp_write	8
multihop	7
phf	4
perl	3
spy	2
pod	264
buffer_overflow	30

Tablo 7.2. *Deneylerde kullanılan sınıf tipi ve sayısı*

Sınıf Tipi	**Sayısı**
normal	788
back	364
guess_passwd	50
imap	16
ipsweep	462
neptune	840
nmap	16
portsweep	5
rootkit	10
satan	462
smurf	1976
teardrop	4
warezclient	7

Tablo 7.3 *Deneylerde kullanılan saldırı tipi ve kategorileri*

Saldırı Tipi	**Saldırı Kategorisi**
back	Dos
guess_passwd	R2L
imap	R2L
ipsweep	Probe
neptune	Dos
nmap	Probe
portsweep	Probe
rootkit	U2R
satan	Probe
smurf	Dos

[Tablo 7.3 (Devam) *Deneylerde kullanılan saldırı tipi ve kategorileri* **].**

teardrop	Dos
warezclient	R2L

7.1. Ortalama Mutlak Hata

Ortalama mutlak hata modelin performans değerini gösteren bir indistir. Bu değer ne kadar küçük olursa performans değeri o kadar iyi olur.

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|x_i - y_i\right| \tag{7.1}$$

Denklem 7.1'de x_i gerçek sınıftaki örnek sayısını y_i doğru tahmin edilen sınıftaki örnek sayısını ifade ediyor. n ifadesi her sınıftaki toplam örnek sayısını ifade ediyor.

7.2. Kök Ortalama Kare Hata

$$\text{RMSE}=\sqrt{\frac{(x_1-y_1)^2+(x_2-y_2)^2+.....+(x_{n-}y_n)^2}{n}} \tag{7.2}$$

Öngörü başarısını ölçmek için kullanılan bir kriterdir. Denklem 7.2'de x_i gerçek sınıftaki örnek sayısını y_i doğru tahmin edilen sınıftaki örnek sayısını ifade ediyor. n ifadesi her sınıftaki toplam örnek sayısını ifade ediyor.

Tablo 7.4. *Karışıklık Matrisi*

Gerçek Sınıf	**Öngörülen Sınıf**	
	Pozitif	Negatif
Pozitif	TP	FN
Negatif	FP	TN

Tablo 7.4'te karışıklık matrisinde satırlar test kümesindeki örneklere ait gerçek değerleri, kolonlar ise modelin tahminini ifade etmektedir.Tablo 7.4'te gösterilen TP ifadesi doğru sınıflandırılmış pozitif örnek sayısı, TN ifadesi doğru sınıflandırılmış negatif örnek sayısı, FN yanlışlıkla negatif olarak sınıflandırılmış pozitif örnek sayısı, FP

yanlışlıkla pozitif olarak sınıflandırılmış negatif örnek sayısını göstermektedir(Modi, Jain, 2015).

Doğru sınıflandırılan örneklerin sayısı: TP+TN (7.3)

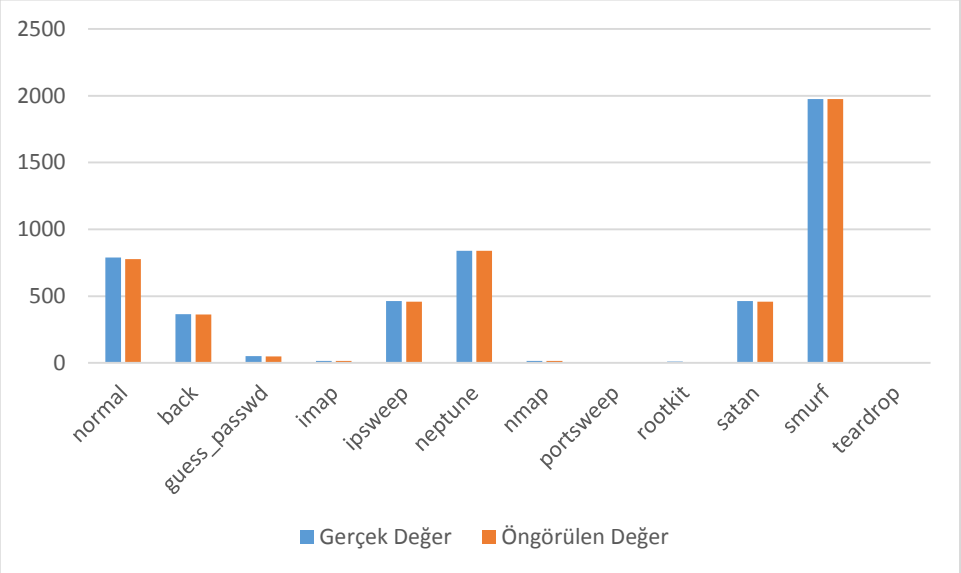
Doğru sınıflandırılan örneklerin yüzdesi= $\frac{TP+TN}{TP+TN+FP+FN}$ (7.4)

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre algoritmaların gösterdiği performanslar Tablo 7.5 ile gösterilmiştir.

Tablo 7.5. *Bütün özelliklere göre algoritmaların karşılaştırılması*

Algoritma Adı	**Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi**	**Ortalama Mutlak Hata**	**Kök Ortalama Kare Hata**
FURIA	4965 - %99.3	0.0006	0.0215
C4.5(J48)	4950 - % 99	0.001	0.028
SVM	4856 - %97.12	0.0025	0.05
MLP	3604 - %72.0	0.0293	0.1199

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre FURIA algoritmasının gösterdiği performansın grafiksel olarak gerçek değer ve öngörülen değerlerin durumları Şekil 7.2 ile ifade edilmiştir.



Şekil 7.2. *Bütün özelliklere göre FURIA algoritması için gerçek değer ve öngörülen değerlerin dağılım grafiği*

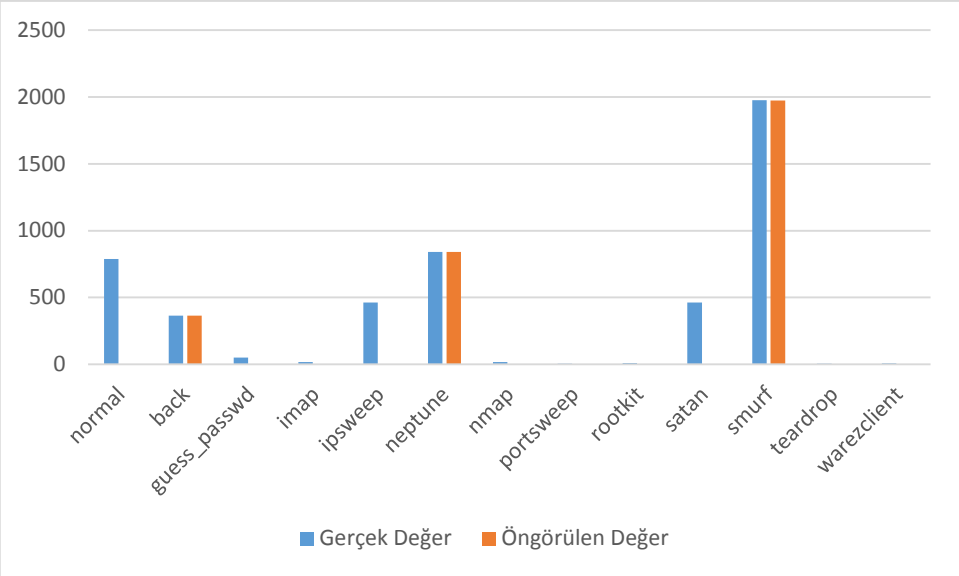
Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre FURIA algoritmasının gösterdiği performansın karışıklık matrisi üzerindeki durumu Tablo 7.6 ile ifade edilmiştir.

Tablo 7.6. *Bütün özelliklere göre FURIA algoritması için karışıklık matrisi*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	778	1	0	0	3	0	0	0	1	3	1	0	1
b=back	1	362	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	1	0	48	0	1	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	3	0	0	0	458	0	0	0	0	0	1	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	1	0	0	0	0	0	0	4	0	0	0	0	0
i=rootkit	3	0	1	0	2	0	0	0	3	0	0	0	1
j=satan	2	0	0	0	1	0	0	0	0	459	0	0	0
k=smurf	0	1	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	1	0	0	0	0	0	0	0	0	0	0	3	0
m=warezclient	4	0	0	0	0	0	0	0	0	0	0	0	3

Tablo 7.6'da karışıklık matrisi temel alınarak gerçekte var olan durumlar ile FURIA algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin back saldırısı 362 tanesi doğru tahmin edilmiş 2 tanesi yanlış tahmin edilmiştir.

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre C4.5 algoritmasının gösterdiği performansın grafiksel olarak gerçek değer ve öngörülen değerlerin durumları Şekil 7.3 ile ifade edilmiştir.



Şekil 7.3. *Bütün özelliklere göre C4.5 algoritması için gerçek değer ve öngörülen değerlerin dağılım grafiği*

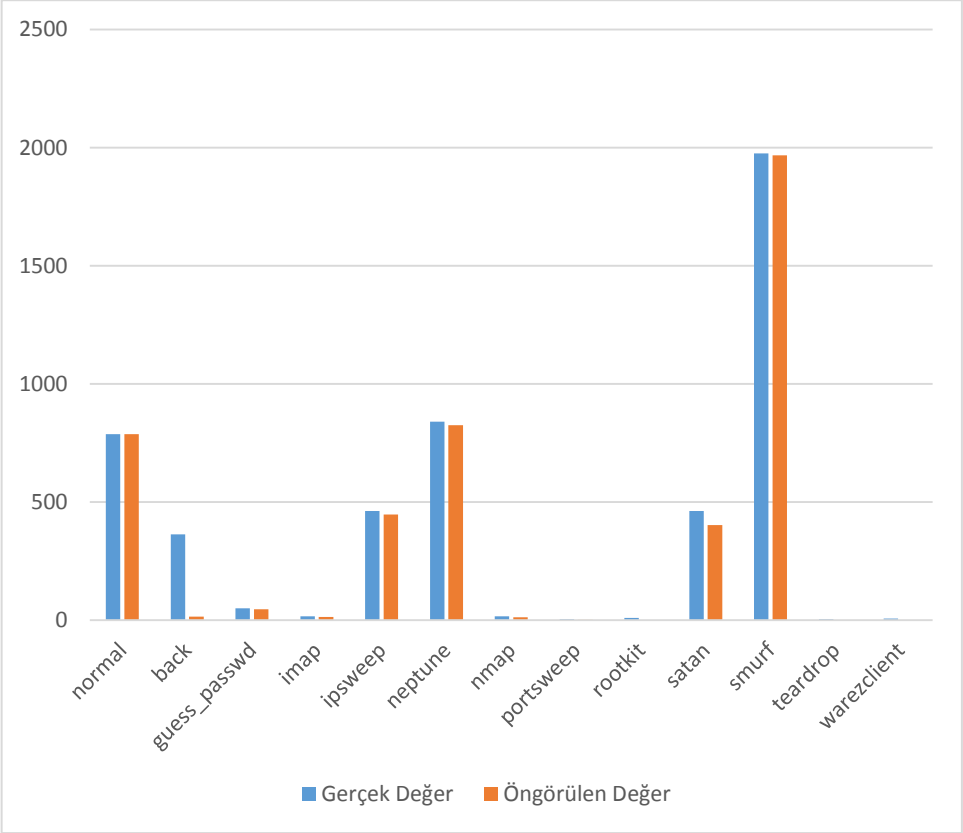
Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre C4.5 algoritmasının gösterdiği performansın karışıklık matrisi üzerindeki durumu Tablo 7.7 ile ifade edilmiştir.

Tablo 7.7. *Bütün özelliklere göre C4.5 algoritması için karışıklık matrisi*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	772	0	1	0	3	0	0	1	2	4	1	1	3
b=back	3	360	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	2	0	48	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	4	0	0	0	457	0	0	1	0	0	0	0	0
f=neptune	0	0	0	0	0	839	1	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	1	0	0	0	1	0	0	3	0	0	0	0	0
i=rootkit	6	0	0	0	3	0	0	0	1	0	0	0	0
j=satan	4	0	0	0	0	0	0	1	0	457	0	0	0
k=smurf	1	0	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	2	1	0	0	0	0	0	0	2	0	0	0	2

Tablo 7.7'de karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *C4.5* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin guess_passwd saldırısı 48 tanesi doğru tahmin edilmiş 2 tanesi yanlış tahmin edilmiştir.

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre SVM algoritmasının gösterdiği performansın grafiksel olarak gerçek değer ve öngörülen değerlerin durumları Şekil 7.4 ile ifade edilmiştir.



Şekil 7.4. *Bütün özelliklere göre SVM algoritması için gerçek değer ve öngörülen değerlerin dağılım grafiği*

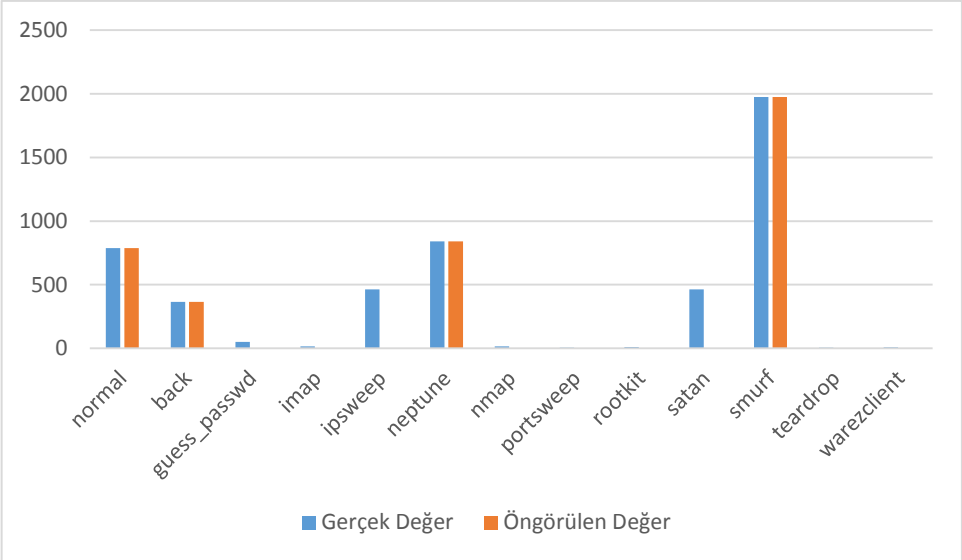
Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre SVM algoritmasının gösterdiği performansın karışıklık matrisi üzerindeki durumu Tablo 7.8 ile ifade edilmiştir.

Tablo 7.8. *Bütün özelliklere göre SVM algoritması için karışıklık matrisi*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	15	349	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	4	0	46	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	14	0	0	0	0	0	0	0	0	0
e=ipsweep	14	0	0	0	448	0	0	0	0	0	0	0	0
f=neptune	5	0	0	0	0	826	0	0	0	9	0	0	0
g=nmap	3	0	0	0	0	0	13	0	0	0	0	0	0
h=portsweep	1	0	0	0	0	0	0	1	0	3	0	0	0
i=rootkit	8	0	0	0	2	0	0	0	0	0	0	0	0
j=satan	37	0	0	0	0	21	1	0	0	403	0	0	0
k=smurf	8	0	0	0	0	0	0	0	0	0	1968	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	7	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 7.8'de karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *SVM* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin ipsweep saldırısı 448 tanesi doğru tahmin edilmiş 14 tanesi yanlış tahmin edilmiştir.

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre MLP algoritmasının gösterdiği performansın grafiksel olarak gerçek değer ve öngörülen değerlerin durumları Şekil 7.5 ile ifade edilmiştir.



Şekil 7.5. *Bütün özelliklere göre MLP algoritması için gerçek değer ve öngörülen değerlerin dağılım grafiği*

Deneylerde kullanılan veri setinde özellik seçimi yapılmadan önce bütün özelliklere göre MLP algoritmasının gösterdiği performansın karışıklık matrisi üzerindeki durumu Tablo 7.9 ile ifade edilmiştir.

Tablo 7.9. *Bütün özelliklere göre MLP algoritması için karışıklık matrisi*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	0	364	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	49	0	0	0	0	1	0	0	0	0	0	0	0
d=imap	16	0	0	0	0	0	0	0	0	0	0	0	0
e=ipsweep	462	0	0	0	0	0	0	0	0	0	0	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	16	0	0	0	0	0	0	0	0	0	0	0	0
h=portsweep	5	0	0	0	0	0	0	0	0	0	0	0	0
i=rootkit	10	0	0	0	0	0	0	0	0	0	0	0	0
j=satan	462	0	0	0	0	0	0	0	0	0	0	0	0
k=smurf	0	0	0	0	0	0	0	0	0	0	1976	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	6	0	0	0	0	0	0	0	0	0	1	0	0

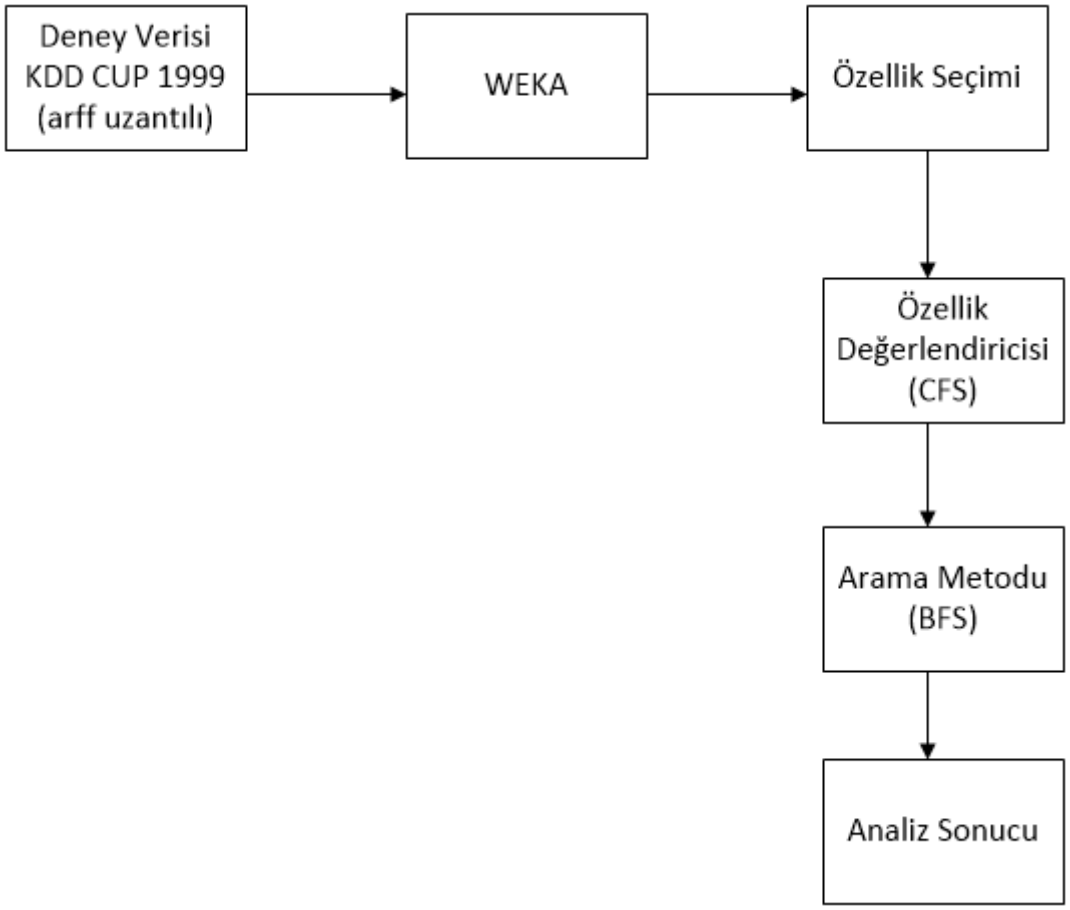
Tablo 7.9'de karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *MLP* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin nmap saldırısı hepsi doğru tahmin edilmiştir.

7.3. WEKA'da Özellik Seçimi

Veri kümesi üzerinde özellik seçme ayarlama işleminin yapıldığı bölümdür. Deneylerde özellik seçimi bölümünde özellik değerlendiricisi olarak CFS(Correlation-based Feature Selection) arama metodu olarak best first search kullanıldı. CFS metodunda öznitelikler arasındaki korelasyon dikkate alınır (Mukherjeea, Sharma, 2012). Veri setlerinin içerdiği gereksiz bilgiyi ayıklamak, kurulan modelin daha anlamlı olmasını sağlamak, modeli hızlandırmak açısından özellik seçimi iyidir(Singh, Kumar, Singla, 2013). Normalde veri setinde 41 tane özellik vardır. Bu özellik kümesinin 2^41 (2,199,023,255,552) tane alt kümesi mevcuttur. Özellik seçimi yaparak en önemli 11 tane özellik seçilmiştir. Özellik seçimi sonucu oluşan bu özelliklerden 11 tane alt küme yaparak karşılaştırma yapıldı. Özellik seçimi yapmadan önce hangi özelliğin daha uygun olduğu bilinmiyor ve dolaysıyla özellik seçimi bu konuda bize yardımcı olmaktadır.

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{7.5}$$

7.5 denkleminde belirtilen CFS için kullanılan M_s ifadesi S öznitelik alt kümesinin sezgisel durumu, k öznitelik sayısını, $\overline{r_{cf}}$ öznitelik ve sınıf değişkenleri arasındaki korelasyonun ortalamasını, $\overline{r_{ff}}$ ifadesi ortalama öznitelikler arasındaki korelasyonu ifade etmektedir (Mukherjee, Sharma, 2012). Özellik seçimi model yapısını anlatan durum Şekil 7.6 ile ifade edilmiştir.



Şekil 7.6. *Özellik Seçimi Model Yapısı*

Şekil 7.6'da belirtildiği gibi WEKA ortamında özellik değerlendirici olarak CFS, arama metodu olarak BFS(Best First Search) algoritması kullanılarak özellik seçimi yapılmıştır.

7.4. En İyi İlk Arama Algoritması

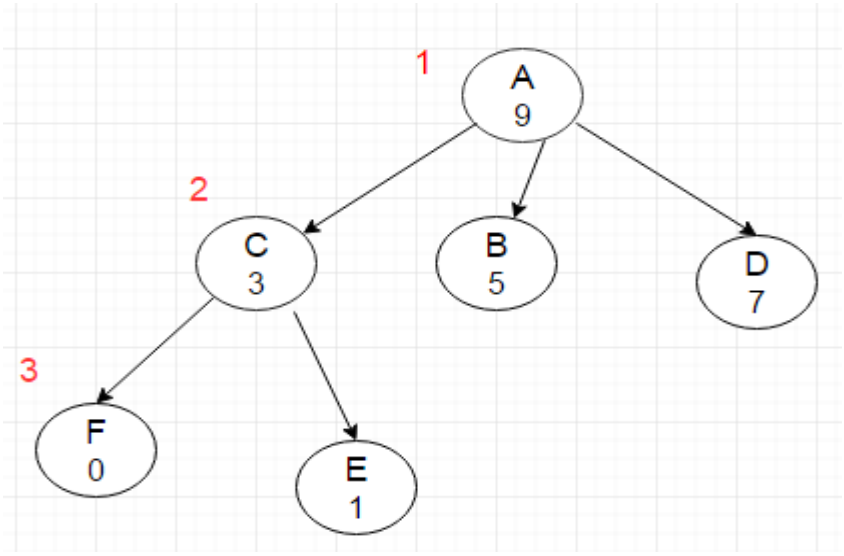
Bu arama algoritmasında aktif olan düğüm genişletilir ve çocuk düğümler değerlendirilir. Bu çocuk düğümlerden en uygun değere sahip olan düğüm bir sonraki genişletme işleminde kullanılmak üzere seçilir. Bu algoritma gideceği yolu heuristic(öngörüsel) değeri küçük olan düğüme göre belirlenir. Arama işlemi çocuk düğümlerinin hepsi en iyi duruma ulaştığında sona erer. Bu algoritmada öngörüsel değerlerin doğru belirlenmesi büyük önem taşır. Bu algoritmada sezgisel bir fonksiyon olur. Düğümler değerlendirme fonksiyonu f(n) göre genişletilir. Düğümü seçme işlemi h(n) sezgisel fonksiyonuna göre yapılır. Varsayım h(n)=0 olarak kabul edilir. Yol bulma probleminde bu sezgisel fonksiyon kuş uçuşu uzaklık olabilir. Sezgisel fonksiyon değeri gerçek değerden küçük olmalıdır. Buradaki algoritma açgözlü BFS olarak geçiyor (http-3).

$$F(n) = h(n) \tag{7.6}$$

$$F(n) = g(n) \tag{7.7}$$

$$h(n) < g(n) \tag{7.8}$$

7.5, 7.6, ve 7.7 denklemlerinde belirtilen F(n) değerlendirme fonksiyonu, g(n) gerçek değerleri, h(n) sezgisel değerleri ifade ediyor. Açgözlü BFS algoritmasının mantığını anlatan durum Şekil 7.7 ile ifade edilmiştir.



Şekil 7.7. *Açgözlü En İyi İlk Arama Algoritması*

Şekil 7.7'de görüldüğü gibi sezgisel değerler bulunmaktadır. Algoritma bu sezgisel değeri en küçük olanı seçerek ilerler böylece en iyi seçeneği bulmaya çalışır. Özellik seçimi model yapısı sonucu elde edilen özelliklerin önem derecesine göre durumu Tablo 7.10 ile ifade edilmiştir.

Tablo 7.10. *BFS Algoritması kullanılarak seçilen özellikler*

Numara	Öznitelik Adı	Tipi
3	service	Ayrık
5	src_bytes	Sürekli
6	dst_bytes	Sürekli
8	wrong_fragment	Sürekli
23	count	Sürekli
24	srv_count	Sürekli
30	diff_srv_rate	Sürekli
35	dst_host_diff_srv_rate	Sürekli

[Tablo 7.10.(Devam) *BFS Algoritması kullanılarak seçilen özellikler* **].**

36	dst_host_same_src_port_rate	Sürekli
37	dst_host_srv_diff_host_rate	Sürekli
40	dst_host_rerror_rate	Sürekli

Tablo 7.10'da belirtilen numara kolonu özelliklerin KDD99 veri setindeki sırasıdır. Öznitelik adları BFS algoritmasındaki önem sırasına göre yukarıdan aşağıya sıralanmıştır. Seçilen özelliklerden service özelliğine göre algoritmaların gösterdiği performans durumları Tablo 7.11 ile ifade edilmiştir.

Tablo 7.11. *Service özelliğine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
C4.5(J48)	4284 - %85.68	0.0175	0.094
SVM	4266 - %85.32	0.0128	0.113
FURIA	4249-%84.98	0.0129	0.1117
MLP	4104- %82.08	0.022	0.104

Seçilen özelliklerden service ve src_bytes özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.12 ile ifade edilmiştir.

Tablo 7.12. *Service ve src_bytes özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4826 - %96.52	0.0033	0.0542
C4.5(J48)	4818 - %96.36	0.0049	0.052
SVM	4817-%96.34	0.0032	0.0564
MLP	4058- %81.16	0.0229	0.1085

Seçilen özelliklerden service, src_bytes, dst_bytes özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.13 ile ifade edilmiştir.

Tablo 7.13. *Service, src_bytes,* dst_bytes *özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4834-%96.68	0.0031	0.053
SVM	4825-%96.5	0.003	0.0552
C4.5(J48)	4818 - %96.36	0.0049	0.0517
MLP	4150 - %83	0.0213	0.1039

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.14 ile ifade edilmiştir.

Tablo 7.14. *Service, src_bytes, dst_bytes, wrong_fragment özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4832 - %96.64	0.0031	0.0533
SVM	4825 - %96.5	0.003	0.0552
C4.5(J48)	4818 - %96.36	0.0049	0.0517
MLP	4160 - %83.2	0.0217	0.1034

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.15 ile ifade edilmiştir.

Tablo 7.15. *Service, src_bytes,* dst_bytes, wrong_fragment, count *özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4911 - %98.22	0.0015	0.0361
C4.5(J48)	4890 - %97.8	0.0025	0.0396
SVM	4860 - %97.2	0.0024	0.0493
MLP	4371 - %87.42	0.0169	0.0903

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.16 ile ifade edilmiştir.

Tablo 7.16. *Service, src_bytes,* dst_bytes, wrong_fragment, count, srv_count *özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4917 - %98.34	0.0016	0.0356
C4.5(J48)	4888 - %97.76	0.0023	0.0386
SVM	4864-%97.28	0.0024	0.0486
MLP	4279- %85.58	0.0184	0.0945

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.17 ile ifade edilmiştir.

Tablo 7.17. *Service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4951 - %99.02	0.001	0.0282
C4.5(J48)	4910 - %98.2	0.0018	0.0347
SVM	4867 - %97.34	0.0023	0.0481
MLP	3717 - %74.36	0.0271	0.1156

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.18 ile ifade edilmiştir.

Tablo 7.18. *Service, src_bytes,* dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate *özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4974 - %99.48	0.0006	0.0204
C4.5(J48)	4931 - %98.62	0.0014	0.0314
SVM	4872-%97.44	0.0022	0.0472
MLP	3499- %69.98	0.0304	0.1253

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.19 ile ifade edilmiştir.

Tablo 7.19. *Service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate özelliklerine göre sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4979-%99.58	0.0005	0.0178
C4.5(J48)	4938 - %98.76	0.0012	0.0297
SVM	4872- %97.44	0.0022	0.0472
MLP	3971- %79.42	0.0232	0.1063

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, *diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliklerine göre algoritmaların gösterdiği performans* durumları Tablo 7.20 ile ifade edilmiştir.

Tablo 7.20. *Service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate dst_host_srv_diff_host_rate özelliklerine göre sınıflandırma sounucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4973 - %99.46	0.0005	0.0188
C4.5(J48)	4962 - %99.24	0.0007	0.0239
SVM	4872 - %97.44	0.0022	0.0472
MLP	4130 - %82.6	0.0209	0.0986

Seçilen özelliklerden service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, *dst_host_srv_diff_host_rate, dst_host_rerror_rate* özelliklerine göre algoritmaların gösterdiği performans durumları Tablo 7.21 ile ifade edilmiştir.

Tablo 7.21. *Service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_rerror_rate özelliklerine göre özelliklerine göre sınıflanırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4975 - %99.5	0.0005	0.0184
C4.5(J48)	4945 - %98.9	0.0013	0.0296
SVM	4872 -%97.44	0.0022	0.0472
MLP	4262 - %85.24	0.0186	0.0945

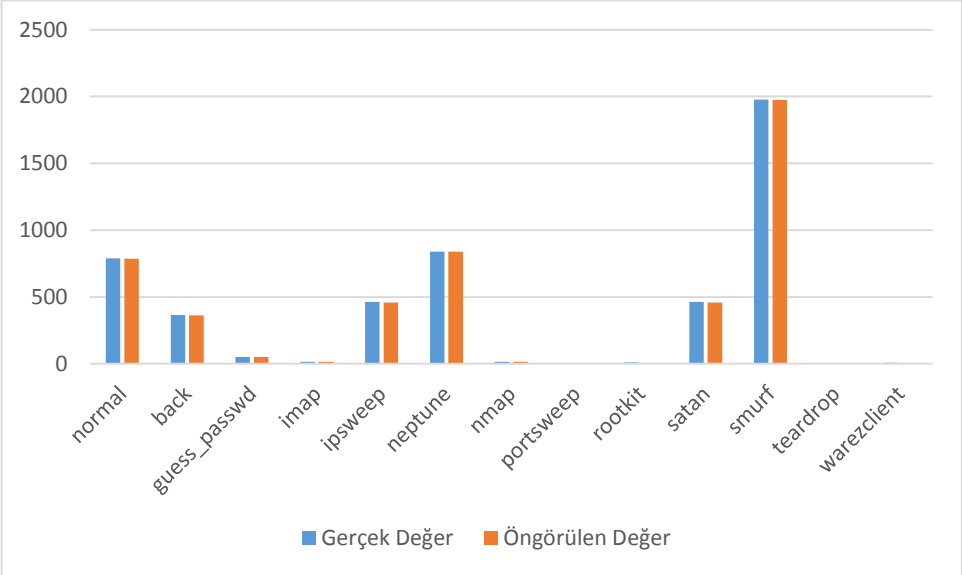
Özellik seçimi sonucu her algoritmanın en iyi olduğu durum Tablo 7.22'de ifade edilmiştir.

Tablo 7.22. *Her algoritmanın en iyi olduğu sınıflandırma sonucu*

Algoritma Adı	Doğru Sınıflandırılan Örneklerin Sayısı-Yüzdesi	Ortalama Mutlak Hata	Kök Ortalama Kare Hata
FURIA	4979-%99.58	0.0005	0.0178
C4.5(J48)	4962 - %99.24	0.0007	0.0239
SVM	4872 -%97.44	0.0022	0.0472
MLP	4371- %87.42	0.0169	0.0903

7.22 tablosunda yapılan bütün karşılaştırmalarda her algoritmanın en iyi olduğu sınıflandırma sonucu eklendi. En iyi algoritmanın FURIA algoritması olduğu gözlemlendi.

Özellik seçimi sonucu FURIA algoritmasının en iyi performans gösterdiği *service*, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği Şekil 7.8 ile gösterilmiştir.



Şekil 7.8. *FURIA algoritması için service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği*

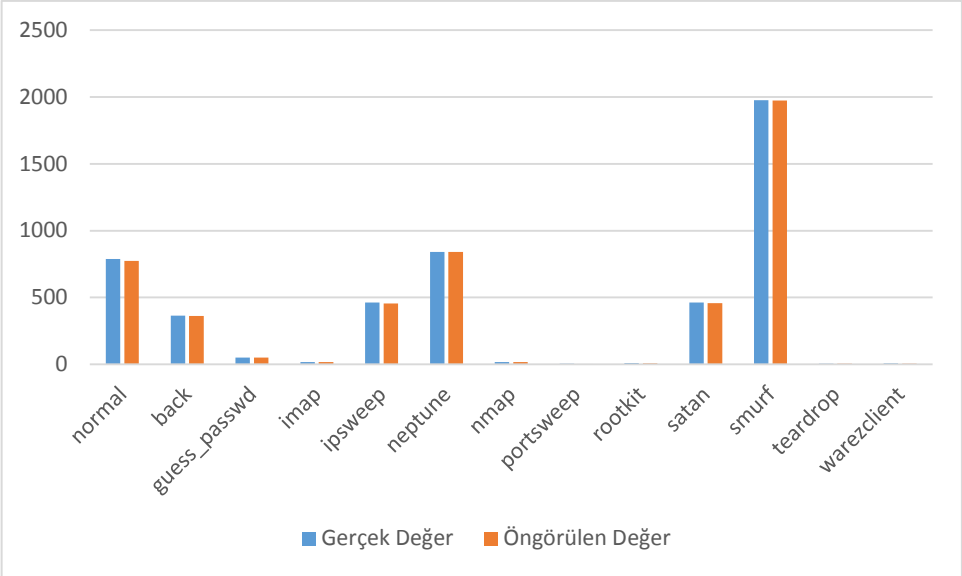
Özellik seçimi sonucu FURIA algoritmasının en iyi performans gösterdiği *service,* src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate özelliklerine göre karışıklık matrisindeki durumu Tablo 7.23'de ifade edilmiştir.

Tablo 7.23. *FURIA algoritması service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, özelliklerine göre karışıklık matrisi sonucu*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	786	0	0	0	0	0	0	0	0	2	0	0	0
b=back	1	363	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	0	0	50	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	1	0	0	0	459	0	0	0	0	0	1	0	1
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	2	0	0	0	0	0	0	0	0	3	0	0	0
i=rootkit	2	0	1	0	1	0	0	0	6	0	0	0	1
j=satan	2	0	0	0	0	1	0	0	0	459	0	0	0
k=smurf	1	0	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	2	0	0	0	0	0	0	0	0	0	0	0	5

Tablo 7.23'de Karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *FURIA* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin rootkit saldırısı 6 tanesi doğru 5 tanesi yanlış tahmin edilmiştir.

Özellik seçimi sonucu J48 algoritmasının en iyi performans gösterdiği service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği Şekil 7.9 ile gösterilmiştir.



Şekil 7.9. *J48 algoritması için service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği*

Özellik seçimi sonucu J48 algoritmasının en iyi performans gösterdiği service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliklerine göre karışıklık matrisindeki durumu Tablo 7.24'de ifade edilmiştir.

Tablo 7.24. *J48 service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliğine göre karışıklık matrisi sonucu*

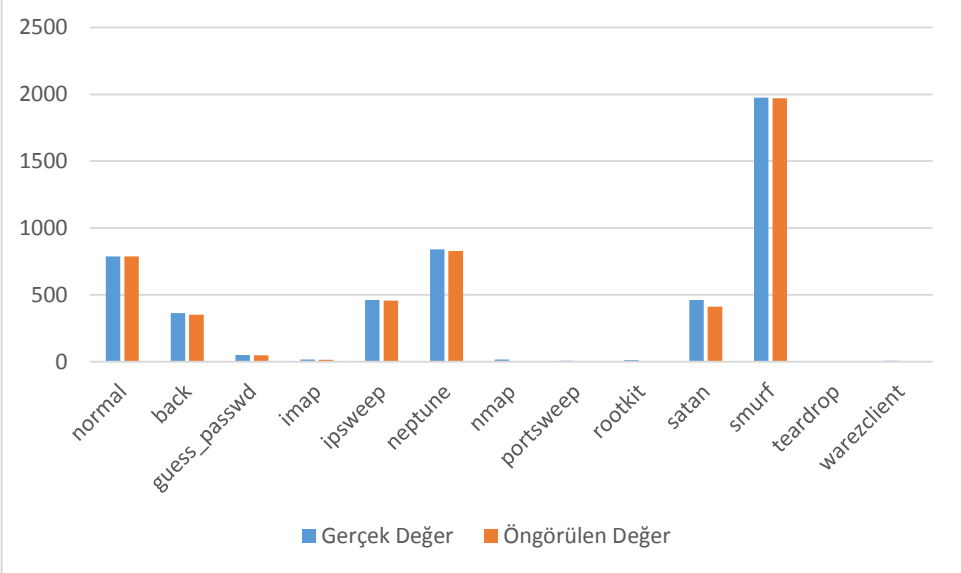
Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	774	2	0	0	2	0	0	0	2	5	1	0	2
b=back	3	361	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	0	0	50	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	2	0	0	0	455	0	0	0	0	0	5	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	0	0	0	0	0	0	1	3	1	0	0	0	0
i=rootkit	1	0	0	0	1	0	0	0	7	0	0	0	1

[Tablo 7.24.(Devam) *J48 service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate özelliğine göre karışıklık matrisi sonucu* ***].***

j=satan	3	0	0	0	0	0	0	1	1	457	0	0	0
k=smurf	0	0	0	0	1	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	1	0	0	0	0	0	0	0	2	0	0	0	4

Tablo 7.24'te Karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *J48* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin satan saldırısı 457 tanesi doğru 5 tanesi yanlış tahmin edilmiştir.

Özellik seçimi sonucu SVM algoritmasının en iyi performans SVM service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği Şekil 7.10 ile gösterilmiştir.



Şekil 7.10. *SVM algoritması için service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği*

Özellik seçimi sonucu SVM algoritmasının en iyi performans gösterdiği *service, src_bytes,* dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate,

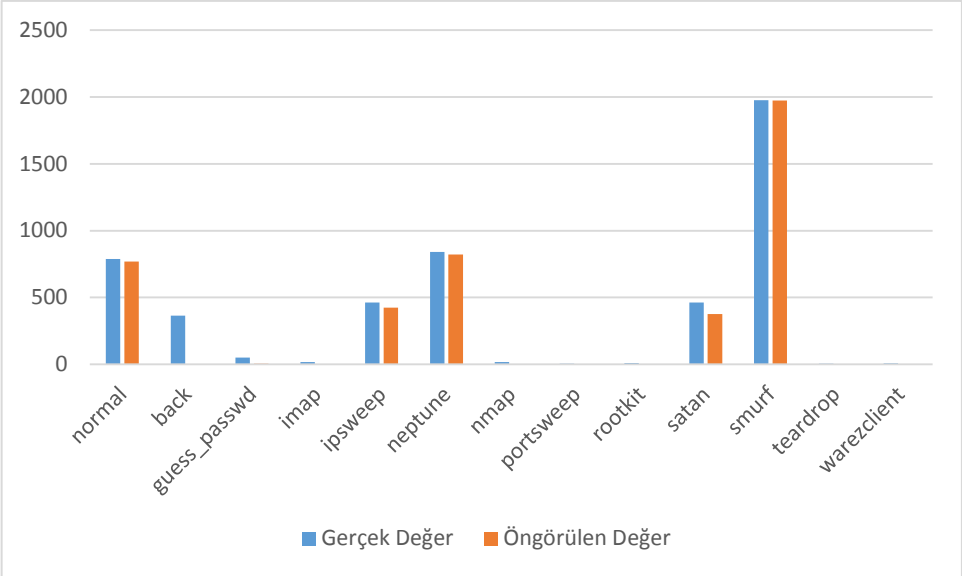
dst_host_diff_srv_rate özelliklerine göre karışıklık matrisindeki durumu Tablo 7.25'te ifade edilmiştir.

Tablo 7.25. *SVM service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate özelliklerine göre karışıklık matrisi sonucu*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	13	351	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	2	0	48	0	0	0	0	0	0	0	0	0	0
d=imap	2	0	0	14	0	0	0	0	0	0	0	0	0
e=ipsweep	5	0	0	0	457	0	0	0	0	0	0	0	0
f=neptune	3	0	0	0	0	828	0	0	0	9	0	0	0
g=nmap	0	0	0	0	16	0	0	0	0	0	0	0	0
h=portsweep	0	0	0	0	5	0	0	0	0	0	0	0	0
i=rootkit	8	0	0	0	2	0	0	0	0	0	0	0	0
j=satan	30	0	0	0	2	20	0	0	0	410	0	0	0
k=smurf	5	0	0	0	0	0	0	0	0	0	1971	0	0
l=teardrop	2	0	0	0	0	0	0	0	0	0	0	2	0
m=warezclient	4	0	0	0	0	0	0	0	0	0	0	0	3

Tablo 7.25'te Karışıklık matrisi temel alınarak gerçekte var olan durumlar ile SVM algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin smurf saldırısı 1971 tanesi doğru 5 tanesi yanlış tahmin edilmiştir.

Özellik seçimi sonucu MLP algoritmasının en iyi performans gösterdiği service, src_bytes, dst_bytes, wrong_fragment, count özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği Şekil 7.11 ile gösterilmiştir.



Şekil 7.11. *MLP algoritması için service, src_bytes, dst_bytes, wrong_fragment, count özelliklerine göre gerçek değer ve öngörülen değerlerin dağılım grafiği*

Özellik seçimi sonucu MLP algoritmasının en iyi performans gösterdiği *service, src_bytes,* dst_bytes, wrong_fragment, count özelliklerine göre karışıklık matrisindeki durumu Tablo 7.26'da ifade edilmiştir.

Tablo 7.26. *MLP service, src_bytes, dst_bytes, wrong_fragment, count özelliğine göre karışıklık matrisi sonucu*

Gerçek	**Öngörülen**												
	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	770	0	1	0	8	9	0	0	0	0	0	0	0
b=back	328	0	0	0	36	0	0	0	0	0	0	0	0
c=guess_passwd	28	0	5	0	12	5	0	0	0	0	0	0	0
d=imap	2	0	2	0	7	5	0	0	0	0	0	0	0
e=ipsweep	34	0	0	0	423	5	0	0	0	0	0	0	0
f=neptune	8	0	0	0	9	821	0	0	0	2	0	0	0
g=nmap	14	0	0	0	0	2	0	0	0	0	0	0	0
h=portsweep	4	0	0	0	0	1	0	0	0	0	0	0	0
i=rootkit	9	0	0	0	0	1	0	0	0	0	0	0	0
j=satan	33	0	1	0	16	35	0	0	0	377	0	0	0
k=smurf	0	0	0	0	1	0	0	0	0	0	1975	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	6	0	0	0	0	0	0	0	0	0	1	0	0

Tablo 7.26'da Karışıklık matrisi temel alınarak gerçekte var olan durumlar ile *MLP* algoritması sonucu tahmin edilen durumlar ifade edilmiştir. Örneğin teardrop saldırısı hepsi yanlış tahmin edilmiştir.

Tablo 7.27'de algoritmaların zaman olarak performansı karşılaştırılmıştır.

Tablo 7.27. *Algoritmaların zaman olarak performansı*

Algoritma Adı	Zaman (sn)
C4.5(J48)	0.8
FURIA	16.7
SVM	56.6
MLP	244.9

8.SONUÇLAR ve ÖNERİLER

Bu tezde günümüzde bilişim sistemlerine yapılan siber saldırılara karşı korunmak için saldırı tespit sistemlerinin geliştirildiğine değinildi. Saldırı tespit sistemlerinin bilişim sistemlerine yapılan bu saldırılara karşı daha etkin bir savunma yapması için kullanılan algoritmalar çok önemlidir. Günümüzde saldırı tespit sistemlerinde değişik algoritmalar kullanılmaktadır. Bu tezde saldırı tespit sistemlerinde kullanılan bazı algoritmalardan olan C4.5, FURIA, MLP, SVM, algoritmaları kullanıldı. Bu algoritmalar WEKA platformunda KDD99 veri setinin bir kısmı kullanılarak performans olarak karşılaştırılmıştır. Yapılan deneylerde ilk önce WEKA'da özellik seçimi yapılmadan bir karşılaştırma yapılmıştır. Bu karşılaştırmalarda temel olarak doğru sınıflandırılan örneklerin sayısı, yüzdesi, ortalama mutlak hata(MAE), kök ortalama kare hata(RMSE) özellikleri referans alınmıştır. Bu karşılaştırmada doğru sınıflandırma açısından en iyi algoritmanın bulanık mantık tabanlı algoritma olan FURIA algoritması olmuştur. Daha sonra özellik seçimi yapılmıştır. Özellik seçimi yapılırken özellik değerlendiricisi olarak CFS seçilmiştir. Arama metodu olarak açgözlü BFS algoritması kullanılmıştır. Elde edilen bu özelliklerden 11 tane karşılaştırma yapılmıştır. Bu karşılaştırmalar sonucunda her bir algoritmanın saldırıyı tespit ederken doğru olanı bulma açısından performans olarak en iyi olduğu sonuçlar tekrar bir tabloya eklenerek karşılaştırma yapılmıştır. Yapılan bu özellik seçimi sonucu her bir algoritma performans açısından daha iyi bir sonuç vermiştir. Özellik seçimi sonucu doğruyu bulma açısından en iyi algoritma yine bulanık mantık tabanlı algoritma olan FURIA algoritması olmuş. En son olarak da bu

algoritmalar aralarında zaman olarak da bir karşılaştırılması yapılmıştır. Bu karşılaştırmada en iyi algoritma C4.5 algoritması olmuştur. Bilişim sistemlerine yapılan saldırı esnasında saldırıyı tespit etmede zaman faktörü de önemli bir etkendir. Diğer algoritmalara göre en düşük performans gösteren algoritma ise çok katmanlı sinir ağlarını ifade eden MLP algoritması olmuştur. Saldırı tespit sistemlerinde doğru saldırıyı tespit etme açısından performans olarak farklılıklar gösteren bu algoritmalar bilişim sistemlerine yapılan saldırılar esnasında çok önem teşkil etmektedir.

KAYNAKÇA

Aggarwala, P.and Sharma, S.K. (2015). Analysis of KDD Dataset Attributes - Class wise For Intrusion Detection. 3rd International Conference on Recent Trends in Computing, 842-851.

Alazab, A., Hobbs, M., Abawajy, J., Alazab, M.(2012). Using Feature selection for intrusion detection system. International Symposium on Communications Information Technologies, 296-301.

Agrawal, S.and Agrawal, J.(2015). Survey on Anomaly Detection using Data Mining Techniques Procedia Computer Science, 708-713.

Amneh, H.A. (2015). Network Intrusion Classification Using Data Mining Studies Techniques. Graduate Thesis. Jordan: Zarqa University, Thesis Faculty of Graduate Zarga University.

A.Aziz, A. S., Hanafi E.S. and Hassanien, A.E. (2017). Comparison of classification techniques applied for network intrusion detection and classification. Journal Applied Logic, 109-118.

Bamakan, S. M. H., Wang, H., Yingjie, T and Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing, 90–102.

Barthakur, P., Dahal, M. and Ghose, M. K. (2015). Adoption of a Fuzzy Based Classification Model for P2P Botnet Detection. International Journal of Network Security,17(5), 522-534.

Belavagi, M. and Muniyal, B.(2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. Procedia Computer Science, 117 – 123.

Canedo, B., Maroño, S. and Betanzos A. (2011). Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. Expert Systems with Applications, 5947–5957.

Çölkesen, R. ve Örencik, B. (2002). Bilgisayar Haberleşmesi ve Ağ Teknolojileri(2). İstanbul:Papatya Yayıncılık.

Dhanabal, L.and Shantharajah, S.P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446-552.

Horng, S.J, Su, M.Y, Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D. (2011) A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications, 306–313.

Hühn J. and Hüllermeier, E. (2005). FURIA: An Algorithm For Unordered Fuzzy Rule Induction, To appear in: Data Mining and Knowledge Discovery.

http-1: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Erişim tarihi: 05.05.2017).

https-2:https://www.ll.mit.edu/ideval/data/1999data.html (Erişim tarihi:5.5.2017)

http-3://www.cs.bilkent.edu.tr/~duygulu/Courses/CS461/Notes/HeuristicSearch.pdf(Erişim tarihi: 10.10.2017).

Kezih,M. and Taibi, M. (2013) .Evaluation Effectiveness of Intrusion Detection System with Reduced Dimension Using Data Mining Classification Tools.2nd. International Conference on Systems and Computer Science (ICSCS), 205-209.

Lin, S., Ying, K., Lee, C and Lee, Z. (2012). An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Applied Soft Computing, 3285–3290.

Manzoor, I. and Kumar, N.(2017). A feature reduced intrusion detection system using ANN classifier. Expert Systems With Applications, 249-257.

Meena, G.and Choudhary, R.R.(2017). A Review Paper on IDS Classification using KDD 99 and NSL KDD Dataset in WEKA. International Conference on Computer, Communications and Electronics, 553-558.

Modi, M. and Jain, A.(2015). A survey of IDS classification using KDD CUP 99 dataset in WEKA. International Journal of Scientific & Engineering Research, 6(11), 947-954.

Mukherjeea, S. and Sharma, N. (2012). Intrusion Detection using Naive Bayes Classifier with Feature Reduction. Procedia Technology, 119 – 128.

Özgür, A. ve Erdem, H. (2012). Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması. BİLİŞİM TEKNOLOJİLERİ DERGİSİ, 5(2), 41-48.

Raman, G., Somu, N., Kirthivasan, K., Liscano, R and Sriram, S. (2017). An efficient intrusion detection system based on hypergraph - Genetic algorithm for paramater optimization and feature selection in support vector machine. Knowledge-Based

Systems, 1–12.

Sahu, S. and Mehtre, B. (2015). Network Intrusion Detection System Using J48 Decision Tree. International Conference on Advances in Computing, Communications and Informatics , 2023-2026.

Singh, S.and Bansal, M. (2013).Improvement of Intrusion Detection System in Data Mining using Neural Network. International Journal of Advanced Research in Computer Science and Software Engineering, 3(9), 1124-1130.

Singh, R., Kumar, H.and Singla, R.K.(2013). Analysis of Feature Selection Techniques for Network Traffic Dataset. International Conference on Machine Intelligence Research and Advancement, 42-46.

Tanrıkulu, H. (2009). Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması. Yüksek Lisans Tezi. Ankara: Ankara Üniversitesi, Fen Bilimleri Enstitüsü.

ÖZGEÇMİŞ

Adı Soyadı: Muhammet Nurullah ÇETER

Yabancı Dil: İngilizce

Doğum Yeri ve Yılı: Cizre/1985

E-Posta: mnceter@anadolu.edu.tr

Eğitim ve Mesleki Geçmişi:

Üniversite: Anadolu Üniversitesi

Fakülte: Mühendislik Fakültesi

Bölüm: Bilgisayar Mühendisliği (2006-2011)

Enstitü : Fen Bilimleri

Anabilim Dalı : Bilgisayar Mühendisliği

Görev : Çözümleyici (2011-)

Kurum : Anadolu Üniversitesi

Birim : Bilgisayar Araştırma ve Uygulama Merkezi