T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMİYLE AĞ ATAKLARININ TESPİTİ

Feyzan SARUHAN ÖZDAĞ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN
Yrd. Doç. Dr. Derya YILTAŞ KAPLAN

II. DANIŞMAN
Yrd. Doç. Dr. Tolga ENSARİ

Haziran, 2017

İSTANBUL

Bu çalışma 22/06/2017 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi:

Yrd.Doç.Dr.Derya YILTAŞ KAPLAN(Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof.Dr.Ahmet SERTBAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi

Doç.Dr.Atakan KURT
İstanbul Üniversitesi
Mühendislik Fakültesi

Yrd.Doç.Dr.Yusuf YASLAN
İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi

Doç.Dr.Rüya ŞAMLI
İstanbul Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Çalışmalarım boyunca değerli yardımlarını esirgemeyen, beni yönlendiren danışmanlarım Yrd. Doç. Dr. Derya YILTAŞ KAPLAN ve Yrd. Doç. Dr. Tolga ENSARİ'ye, öğrenim hayatım süresince beni yüreklendiren, maddi ve manevi desteklerini esirgemeyen sevgili aileme, yüksek lisans eğitimim süresince yardımlarını ve desteğini sürekli hissettiğim sevgili eşim İsmetullah ÖZDAĞ'a anlayış ve destekleri için teşekkür ederim.

Haziran, 2017 Feyzan SARUHAN ÖZDAĞ

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

Ɛ : Eşik değeri
D : Mesafe
A_b : Antikor
A_g : Antijen

Kısaltmalar **Açıklama**

ACK : Acknowledgement
NSA : Negatif Seçim Algoritması
STS : Saldırı Tespit Sistemi
WEKA : Waikato Environment for Knowledge Analysis
PCA : Principal component analysis
DDA : Doğrusal Diskriminant Analizi
TBA : Temel Bileşen Analizi
TB : Temel Bileşen
DOS : Denial of Service
IP : Internet Protocol
SYN : Synchronize
TCP : Transmission Control Protocol
DDOS : Distributed Denial Of Service
ICMP : Internet Control Message Protocol
R2L : Remote to Local
U2R : User to Root
LAN : Local Area Network
GA : Genetik Algoritma
YBS : Yapay Bağışıklık Sistemi
SOM : Self Organizing Maps

ÖZET

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMİYLE AĞ ATAKLARININ TESPİTİ

Feyzan SARUHAN ÖZDAĞ

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Yrd. Doç. Dr. Derya YILTAŞ KAPLAN

II. Danışman : Yrd. Doç. Dr. Tolga ENSARİ

Saldırı Tespit Sistemleri (STS), sürekli gelişen ağ yapıları içerisinde ağ güvenliğini tehdit eden unsurlara karşı kullanılan önemli araçlardan biridir. Gelişen teknolojiyle birlikte güçlü STS'lerin tasarlanması ve bunların ağ sistemleri içerisine entegre edilmesi gereklilik haline gelmiştir. Makine öğrenmesi, bilgisayarlara öğrenme olanağı sağlayan bir yapay zeka türüdür. Analitik model oluşturma işlemi otomatikleştirilerek yeni veri girişleri sonrasında sistemin karar vermesini sağlayacak bilgisayar programlarının geliştirilmesi hedeflenmektedir. Model oluşturma sürecinde, veri aracılığıyla problemi öğrenen algoritmalar kullanılır. Makine öğrenmesi algoritmaları çoğunlukla denetimli öğrenme ve denetimsiz öğrenme olarak ikiye ayrılmaktadır. Denetimli öğrenme algoritmaları, geçmişte öğrendiklerini yeni verilere uygulayarak, denetimsiz öğrenme algoritmaları ise veri kümelerinden çıkarımlar yaparak öğrenme işlemini gerçekleştirmektedir.

Makine öğrenmesi algoritmalarının performanslarının arttırılması için çeşitli yöntemler kullanılmaktadır. Veri kümesi içerisindeki özelliklerin ilişkilerini bularak, ilişkili olmayan verileri analiz dışında bırakmak hem zaman hem de sonuçların doğruluğu açısından olumlu etkiler yaratmaktadır. Bu çalışma kapsamında özellik seçim yöntemi olarak bilgi kazanımı ve özellik çıkarım yöntemi olarak da temel bileşen analizi kullanılarak doğruluk oranları üzerindeki etkileri gözlemlenmiştir.

Saldırı tespiti için makine öğrenmesi algoritmalarından Yapay Bağışıklık Sistemi (YBS) kullanılmıştır. YBS, insan bağışıklık sisteminden esinlenerek oluşturulmuş ve saldırı tespiti için oldukça etkili çalışan bir algoritmadır. YBS'nin öğrenme sürecinde efektifliğini arttırmak için geleneksel detektör üretim tekniklerine ek olarak genetik algoritma kullanılarak hibrid bir çözüm geliştirilmiştir.

YBS ile geliştirilmiş modelin, eğitim ve test aşamalarında KDD Cup 99 veri seti kullanılmıştır. STS'lerle ilgili yapılan çalışmaların çok büyük bir kısmında bu veri kümesi kullanılmaktadır. Çalışma kapsamında geliştirilen sistemin test sonuçlarının yanısıra WEKA aracılığıyla makine öğrenmesi algoritmalarından birkaçı kullanılarak elde edilmiş sonuçlar da paylaşılmıştır. Sonuç olarak temel bileşen sayısının YBS üzerindeki etkileri ve diğer algoritmalardan daha iyi çalıştığı gözlemlenmiştir.

Haziran 2017,75 sayfa.

Anahtar kelimeler: Saldırı Tespiti, Temel Bileşen Analizi, Bilgi Kazanımı, Yapay Bağışıklık Sistemi, Genetik Algoritma

SUMMARY

M.Sc. THESIS

DETECTION OF NETWORK ATTACKS WITH MACHINE LEARNING METHOD

Feyzan SARUHAN ÖZDAĞ

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Derya YILTAŞ KAPLAN

Co-Supervisor : Asst. Prof. Dr. Tolga ENSARİ

Intrusion detection systems are one of the most important tools used against the threats to network security in ever-evolving network structures. Along with evolving technology, it has become a necessity to design powerful intrusion detection systems and integrate them into network systems. Machine learning is a type of artificial intelligence that enables computers to learn. It is aimed to develop the computer programs that will enable the system to decide after the new data entry by automating the analytical model building process. In the modeling process, algorithms that learn the problem through data are used. Machine learning algorithms are mostly divided into supervised and unsupervised learning. Supervised algorithms perform learning process by applying what they learned in the past to new data and unsupervised algorithms by making inferences from datasets.

Different methods are used to improve the performance of machine learning algorithms. Finding relationships among features in the dataset and excluding non-related data from analysis creates positive effects both in terms of time and accuracy of results. Within the scope of this study, effects on accuracy rates were observed by using information gain as feature selection method and principal component analysis as feature extraction method.

Artificial Immune System was used as machine learning algorithm for intrusion detection. The artificial immune system is an algorithm that is inspired by the human immune system and works very efficiently for intrusion detection. In order to increase the effectiveness of the artificial immune system learning process, a hybrid solution has

been developed using genetic algorithm in addition to traditional detector production techniques.

KDD CUP'99 dataset was used in the training and test phases of the model developed with artificial immune system. Most of the studies on intrusion detection systems use this dataset. In addition to the test results of the system developed in the study, the results obtained by using a few of the machine learning algorithms via WEKA are also shared. As a result, it has been observed that the effect of principal components numbers on the artificial immune system algorithm and the artificial immune system algorithm works better than the other algorithms.

June 2017, 75 pages.

Keywords: Intrusion Detection, Principal Component Analysis, Information Gain, Artificial Immune System, Genetic Algorithm

1. GİRİŞ

Günümüzde bilişim çağının etkisiyle artan internet ve ağ teknolojileri kullanımı güvenlik konusunu önemli bir hale getirmiştir. Artan kullanıcı sayısı ve çeşitlilik gösteren kullanıcı profiliyle birlikte ağ üzerinde gerçekleştirilen saldırılar yaygın hale gelmiştir. Bu güvenlik tehditleri karşısında bilginin erişilebilirliğinin, bütünlüğünün ve gizliliğinin korunabilmesi için kurumların iç yapılarında çeşitli güvenlik politikaları geliştirilmeye başlanmıştır. Bu politikalar ağın ayrılmaz bir parçası haline gelen çeşitli güvenlik uygulamalarıyla desteklenmektedir. Yazılımsal ve donanımsal olabilen bu elemanlara örnek olarak güvenlik duvarları ve bu tez kapsamında ele alınacak saldırı tespit sistemleri (STS'ler) gösterilebilir.

STS'lerin ağ üzerinde gerçekleşen aktivitelerin saldırı olup olmadıklarını anlayabilmeleri için bir öğrenme sürecinden geçirilmesi gerekmektedir. Makine öğrenmesi insanın öğrenme yöntemlerini baz alarak makineleri eğiten ve kendi kendilerine karar verebilme yeteneğini kazandırmaya çalışan bir bilim alanıdır. Bu yöntem aracılığıyla bir STS geliştirilerek ağ üzerindeki saldırıların saptanması sağlanacaktır.

Ağ saldırılarının tespit edilmesi için farklı makine öğrenmesi yöntemlerinin kullanıldığı çok sayıda çalışma mevcuttur. Kullanılan yöntemlerden bazıları şunlardır;

- Karar ağaçları [1]
- Destek Vektör Makinesi [2]
- Yapay Sinir Ağları [3]
- Farklı hibrid yöntemler [4, 5]

Bu tez kapsamında ağ saldırılarını saptamak için yapay bağışıklık sistemi (YBS) algoritmaları kullanılacaktır.

YBS insan bağışıklık sisteminden esinlenerek oluşturulmuş bir sistemdir. 1990'lı yılların sonlarına doğru ivme kazanmaya başlamış ve özellikle virüs tespiti, dolandırıcılık tespiti ve arıza kontrolü [6] gibi konularda entegrasyonu sağlanmış etkili bir yöntem olmuştur. Tez kapsamında bu yöntemin kullanılma sebebi negatif seçim algoritmalarının saldırıları sınıflandırmaya oldukça uygun olmasıdır. Negatif seçim

algoritmaları (NSA), doğal bağışıklık sistemi içerisinde yer alan antijen ve antikor yapısını kullanmaktadır. Ağ içerisindeki saldırılar antijen olarak düşünüldüğünde, öğrenme süreci içerisinde tanınan öz hücreler antikor olarak adlandırılmaktadır. NSA ağ içerisindeki hareketleri daha önce belirlediği detektörler ile karşılaştırarak bu hareketlerin saldırı olup olmadıklarına karar vermektedir.

Ağ trafiğinin izlenerek loglanması hem maliyetli olabileceği, hem de kurumların ağ trafik bilgilerini paylaşmak istememelerinden dolayı bu tarz işlemler için kullanılan hazır veri setleri oluşturulmuştur. Bu çalışma kapsamında STS'ler ile ilgili yapılan akademik çalışmalarda en sık kullanılan veri seti olan KDD Cup 99 kullanılmıştır. DARPA 1998 ve DARPA 1999 veri setlerinin başka bir versiyonu olan bu veri seti, Amerikan Hava Kuvvetlerinin ağ yağısına benzetilmeye çalışılarak MIT Lincoln Laboratuvarlarında hazırlanmış bir veri setidir.

Büyük veri setleri içerisinden anlamlı verinin çıkarılması ve daha etkili sonuçların elde edilmesi için kullanılan farklı yöntemler mevcuttur. Bu çalışma kapsamında temel bileşen analizi (TBA) ve bilgi kazanımı yöntemleri kullanılmıştır. TBA bir boyut indirgeme işlemi olup daha etkili sonuçlar alınmasını sağlayan bir yöntem olduğu için tercih edilmiştir.

Tezin genel kısımlar bölümünde ağ ve ağ güvenliği, STS'ler ve hangi tür saldırıların ele alınacağı ile ilgili bilgiler, makine öğrenmesi algoritmaları, tez kapsamında ele aldığımız YBS algoritmalarına ve genetik algoritmalara yer verilmiştir. Bu bölümde aynı zamanda YBS'nin çıkış noktası olan doğal bağışıklık sistemine de yer verilmiştir. Yine bu bölüm kapsamında büyük veri kümeleriyle çalışmanın yarattığı bazı olumsuzlukların ortadan kaldırıldığı yöntemler olan özellik seçimi ve boyut indirgeme tekniklerine yer verilmiştir.

3. Bölümde kullanılan veri seti ve araçlar ile bilgilere yer verilirken, 4. Bölümde ise uygulama kısmına yer verilmiştir. 5. Bölüm olan tartışma ve sonuç kısmında ise çalışmanın çıktıları değerlendirilmiştir.

2. GENEL KISIMLAR

2.1. AĞ VE AĞ GÜVENLİĞİ

Dünya üzerindeki iletişim ağı her geçen gün internetin ve yeni teknolojilerin gelişmesiyle beraber aynı doğrultuda gelişim göstermektedir. Artan kullanıcı sayısı ve değişen kullanıcı profiliyle birlikte ağ üzerindeki güvenlik konusu büyük bir önem kazanmıştır.

Ağ güvenliği, ağ için oluşturulan altyapıya izinsiz erişimleri engelleme ve veri bütünlüğünü, gizliliğini çeşitli saldırılara karşı koruma işlemidir. Ağ güvenliği yönetimi, ağ üzerindeki bileşenleri ve bilgiyi korumak için bir takım politikalar, prosedürler ve uygulamalardan oluşur. STS'ler ve güvenlik açığını saptamak için kullanılan bir takım uygulamalar güvenlik yönetiminin bir parçasıdır. Kurumlar belli prosedürler çerçevesinde güvenlik alt yapılarında farklı uygulamaları barındırır. Bunlar STS'ler, Antivirüs programları, Güvenlik Duvarları, kriptoloji vb. uygulamalardır.

Güvenlik Duvarları aracılığıyla ağ içerisine hangi servislerin erişebilecekleri ile ilgili ağ güvenlik politikaları belirlenir. Güvenlik Duvarları izinsiz girişleri engelleyebilir ancak ağın içerisine girebilecek zararlı yazılımları engellemek konusunda yetersiz kalmaktadır [7]. Bu noktada Antivirüs yazılımları veya STS'ler kötü amaçlı yazılımları tespit etmeye çalışmaktadır [8].

Güvenli bir ağ oluşturmak için dikkat edilmesi gereken noktalar aşağıdaki gibidir [9].

- Erişilebilirlik
- Gizlilik
- Kimlik Doğrulama
- Bütünlük

2.2. SALDIRI TESPİT SİSTEMLERİ

2.2.1. Saldırı

2.2.1.1. Saldırı Tanımı

Günümüzde ağ sistemlerinin yaygınlaşmasıyla birlikte kullanım alanları ve kullanıcı sayısı gittikçe artmaktadır. Bu artış oranı, beraberinde artan saldırı tehditlerini de getirmektedir. Özellikle internet kullanan kullanıcı profili oldukça farklılık gösterdiğinden bilginin gizliliğini ve bütünlüğünü korumak önem kazanmaktadır.

Saldırı, ağ sistemlerine çeşitli zararlar vermek için yapılan girişimler olarak tanımlanabilir. Bu girişimler sistemlere zarar vererek iletişimi engellemek, verilere ulaşarak bilgi sızıntısına neden olmak ve aynı şekilde veriye eriştikten sonra veri üzerinde değişiklikler yaparak bütünlüğü bozmak gibi amaçlara dayanabilmektedir. Bu noktada STS'ler ağ üzerindeki normal ve anormal davranışları birbirinden ayırarak anormal davranışları sistem yöneticilerine haber vermekle görevlidir.

2.2.1.2. Saldırı Türleri

Saldırı türleri 4'e ayrılmaktadır. Bunlar aşağıdaki gibidir.

- **Bilgi Tarama**: Belirli bir sunucu üzerindeki IP'leri, portları veya işletim sistemini öğrenmek için yapılan saldırılardır [10]. Genellikle sistemin zayıf noktalarını veya açıklarını daha sonra kullanılmak üzere tespit ederek sistem güvenliğini aşmak için kullanılan yöntemlerdir [11]. Bu saldırı iki şekilde yapılabilmektedir. Bunlar;
  - **Ipsweep:** Sunucu üzerindeki sadece bir portun sürekli taranması işlemidir.
  - **Portsweep:** Sunucu üzerindeki hizmetleri bulmak için tüm portların sürekli taranması işlemidir.

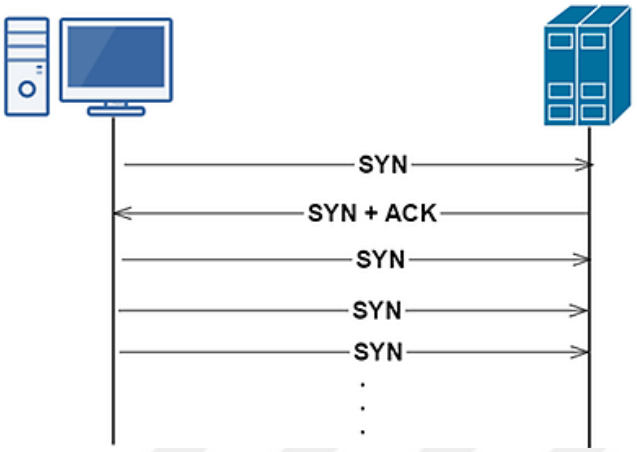
- **Hizmet Engelleme (Denial of Service –DoS):** DoS saldırıları, bir sunucu üzerindeki iş yükünün sürekli istek gönderilerek arttırılması ve bunun sonucunda sunucunun gelen isteklere cevap veremeyecek duruma getirilerek hizmet vermesini engellemeyi amaçlayan bir saldırı türüdür. Sunucular gelen isteklerin gerçek mi, saldırı mı olduğunu anlayamadıkları için saldırıların önüne geçmenin

bir yolu bulunmamaktadır. DoS saldırısının gerçekleştirilmesi için bilgisayarlara virüs bulaştırılarak zombi bilgisayarlar oluşturulur ve zombiler aracılığıyla saldırı yapılır. Güvenlik Duvarları ve Antivirüs programları bu saldırılar karşısında oldukça yetersiz kalmaktadır. Bu sebeplerden dolayı DoS saldırılarını engellemek çoğu zaman mümkün değildir. DoS saldırısından korunmak için paralel sunucu yapısı kullanılabilir. Böylelikle sunuculardan biri saldırıya uğrasa bile diğer sunucular hizmet vermeye devam edebileceklerdir. DoS saldırılarının başlıcaları aşağıdaki gibidir.

- **Ping-of-death:** Ping, ağ üzerindeki başka bir bilgisayara paket göndererek bilgisayarın erişilebilir olup olmadığını anlamaya yarayan bir işlemdir. Gönderilen paketin gidip gelme süresi ping süresi olarak adlandırılır. IP tarafından cihazlara verilen maksimum paket boyutu 65 bayt civarındadır. Saldırı sırasında 65 bayttan daha büyük paketler gönderilerek işletim sisteminin çökmesi hedeflenmektedir.

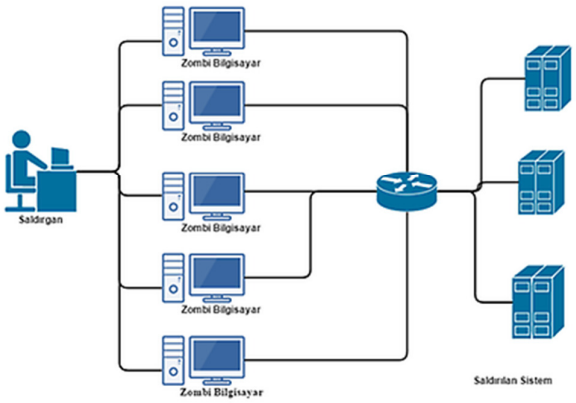
- **Arabellek aşımı:** Arabellek hafızada art arda veri depolamak için kullanılan kısımdır. Bu alana boyutundan daha büyük veri gönderilerek yani çok fazla trafik oluşturularak saldırı gerçekleştirilir. En yaygın olarak kullanılan DoS saldırılarındandır.

- **SYN saldırıları:** SYN, bilgisayarlar arasında TCP bağlantısı kurulurken gönderilen mesajdır. SYN saldırıları, geçersiz veya sahte IP adreslerinden gelen isteklere cevap vermeye çalışırken sunucunun bir süre sonra geçerli isteklere cevap verememesi ve hatta bir süre sonra hiç bir isteğe cevap verememesiyle sonuçlanan bir saldırı türüdür. Sunucu sahte IP'ye, haberleşme sonucunda alınan SYN mesajına karşılık SYN-ACK (onay paketi) paketi yollamaktadır. Gönderdiği pakete karşılık beklediği onay (ACK) mesajını alamadığı için SYN-ACK paketini sürekli göndermeye çalışmaktadır. Saldırı sonucunda sunucu bir süre sonra hizmet veremez duruma gelmektedir.



Şekil 2.1: SYN saldırısı.

- **Dağıtılmış Hizmet Reddi (Distributed Denial Of Service - DDoS):** DoS saldırılarının temel mantığı tek bir kaynaktan saldırı düzenlemeye dayanmaktadır. DDoS'la birlikte saldırı yapan kaynak sayısı artmaktadır. Kaynak sayısını arttırmak için bilgisayarlara zararlı içerikli yazılımlar çeşitli yollarla gönderilmektedir. Bu yazılımların çalışması sonucunda bilgisayarlar ele geçirilerek tek bir kaynağa saldırı başlatılır ve sistemler hizmet veremez duruma getirilmeye çalışılır.



Şekil 2.2: DDoS saldırısı.

- **Teardrop Saldırıları:** Bilgisayarlar internet üzerinden gelen paketleri kendi içinde parçalayarak iletmektedir. Bu parçalama işlemi paketler içerisindeki ofset bilgisine göre yapılır. Dolayısıyla bu ofset bilgilerinin çakışmaması gerekmektedir. Saldırganlar paket içerisinde üstüste gelecek ofsetler ekler. Böylece bu durumu yönetemeyen bilgisayarlar için sistemin çökmesi kaçınılmazdır.

- **Servislere Aşırı Yüklenme:** Belirli bir kullanıcı ve servisi düşürmek için kullanılan saldırı tipidir. Saldırı sırasında kullanıcıya çok sayıda ICMP paketi gönderilmektedir.

- **Yönetici Hesabı ile Yerel Oturum Açma (Remote to Local – R2L):** Bir ağ içerisinde sisteme paketler gönderilerek oluşturulmaya çalışılan ancak paket gönderen kişinin sistem kullanıcısı olmadığı saldırılardır. Amaç, erişim yetkisi olmadığı halde erişim yetkisi olan bir kullanıcı gibi sisteme dahil olarak güvenlik sistemini aşmaktır. Saldırganın sistem üzerinde yerel veya misafir bir kullanıcı hesabını ele geçirmesinin çeşitli yolları bulunmaktadır [12]. Bu saldırı türüne Dictionary, Guest ve Xsnoop örnek olarak gösterilebilir.

  - **Guest:** Bazı sistemlerde misafir hesapları için ya şifre bulunmamaktadır ya da şifre olsa bile kolay tahmin edilebilir olmaktadır. Çoğu işletim sistemi misafir hesabıyla aktive edilerek gönderilmektedir. Bu da saldırganın yararlanmaya çalışacağı ilk açıklardandır. Genel olarak şifrelerin tahmin edilerek sisteme giriş yapılması hedeflenmektedir.

- **Kullanıcı Hesabını Yönetici Hesabına Yükseltme (User to Root – U2R):** Saldırganın sistem üzerinde normal bir kullanıcı hesabını ele geçirmesiyle başlayan saldırı türüdür. Sistem üzerinde yönetici yetkisi olan kullanıcıların yapabilecekleri işleri yapmaya çalışarak gerçekleştirilen bir saldırı türüdür [12].

  - **SqlAttack:** Saldırganın veritabanına erişerek orada saldırı gerçekleştirmesi için komut satırı elde etmesidir.

2.2.2. Saldırı Tespit Sistemleri Türleri

STS'ler ağ üzerindeki olayları analiz eden ve saldırıları tespit ederek ağ yöneticisine haber veren sistemlerdir. Genel olarak amaçları yukarıda bahsedilen saldırı türlerini tespit etmek ve sistemi tehditlerden korumaktır. Sistem güvenliği ve dolayısıyla veri erişilebilirliğini yönetmek isteyen kurumlar güvenlik sistemlerinin bir parçası olarak STS'leri kullanmaktadır. Mevcutta kullanılan çok fazla STS bulunmaktadır. Bu STS'ler farklı gruplar tarafından geliştirildiği gibi farklı yaklaşımlara da dayanabilmektedir. STS'ler 4 ana başlık altında sınıflandırılabilir.

2.2.2.1. Veri İşleme Zamanına Göre

STS'ler veri işleme zamanına göre ikiye ayrılmaktadır. Bunlar;

- **Gerçek Zamanlı (Continuous):** Ağ üzerindeki bilgi akışından anında haberleri olan STS'lerdir. Bu sayede saldırıları hemen analiz ederek kısa bir sürede cevap verme özellikleri mevcuttur.

- **Gerçek Zamanlı Olmayan (Batch Mode):** Gerçek zamanlı STS'lerin aksine ağ üzerindeki bilgi akışından anlık bir şekilde haberleri bulunmamaktadır. Bilgiler önce bir yerde depolanır ve belli periyotlarda analiz edilir. Bu sebepten saldırılara hemen cevap verilememektedir.

2.2.2.2. Mimarı Yapıya Göre

Ağ üzerindeki verileri toplama şekline göre ikiye ayrılmaktadır. Bunların ilki sistem üzerinde farklı noktalardan bilgi toplayarak analiz eden dağıtık sistemler, ikincisi ise tüm verileri tek bir merkezden toplayarak analiz eden sistemlerdir [10].

2.2.2.3. Bilgi Kaynaklarına Göre

Bazı STS'ler saldırıları tespit etmek için LAN segmentlerine yakalanan paketleri analiz ederken bazıları işletim sistemleri veya uygulama yazılımları tarafından saldırı sinyali olarak üretilen bilgi kaynaklarını analiz etmektedir. STS'ler bilgi kaynaklarına göre ikiye ayrılmaktadır. Bunlar;

- **Ağ-temelli STS'ler:** En yaygın olarak kullanılan STS'lerdir. Ağ trafiği içerisindeki paketleri yakalar ve bunları analiz eder. Tüm ağı izlediği için ağ üzerinde herhangi bir bilgisayarda meydana gelen saldırıyı tespit edebilmektedir. Ağ temelli STS'ler tek amaçlı sensörlerden ya da ağ üzerinde farklı konumlandırılmış sunuculardan oluşur [13].

- **Sunucu-temelli STS'ler:** Sunucu temelli STS'ler ilk geliştirilen ve uygulanan sistemlerdir [14]. Genellikle kritik bilgisayarlar ve sunucular üzerine kurulmaktadır [14]. Bu sistemler her bilgisayara ayrı ayrı kurulup sadece kurulu olduğu bilgisayarı takip etmektedir. Saldırılar incelendikleri zaman sadece dışarıdan değil, içeriden gelebildikleri de görülmüştür. Bu yüzden sistem logları incelenerek saldırının kim tarafından ve nasıl yapıldığı anlaşılmaya çalışılır.

2.2.2.4. Saldırı Tespit Yöntemlerine Göre

STS'ler iki saldırı tespit yöntemine göre çalışmaktadır. Bunlar;

- **Anormallik Tespiti:** Kullanıcıların davranışlarını ölçümlemeye ve bunun sonucunda oluşan verilere göre kullanıcı davranışının normal mi, anormal mi olduğuna karar vermeye çalışmaktadır. Genel olarak ağ üzerindeki trafik izlenerek belirlenen eşik değerlerine göre bir değerlendirme yapılmaktadır. Tahmine dayalı bir yöntem olduğu için uzman sistemler veya bulanık mantık gibi sistemlerden faydalanılmaktadır.

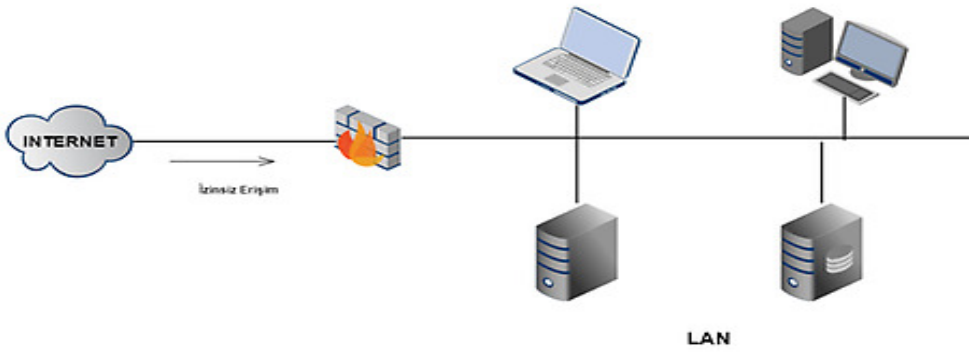
- **İmza-tanıma Temelli Saldırı Tespiti:** Kötüye kullanım tespiti olarak da bilinen bu yöntemde sistemin zayıf noktalarına veya güvenlik politikalarına göre belirlenen imzalar çerçevesinde değerlendirme yapılmaktadır. Sistem üzerinde tanımlı her davranışın bir imzası vardır. Bu imzalar dışında kalan davranışlar normal olarak adlandırılır. Yeni gerçekleşen her saldırı tipi için bir imza belirlenerek bu sistemlere tanıtılmalıdır [7].

2.3. SALDIRI TESPİT SİSTEMLERİ VE GÜVENLİK DUVARLARI

Ağ güvenliği söz konusu olduğunda akla gelen en yaygın uygulama güvenlik duvarlarıdır. Ancak güvenlik duvarının tek başına sistemi koruması mümkün değildir. Güvenlik duvarları belli politikalar çerçevesinde sisteme erişimleri düzenler. Organizasyonlar arasındaki iletişimde kurumlar arası iletişimi düzenlemek için güvenlik duvarları ile ilgili tanımlamalar yapılır. Bu tanımlar sayesinde erişimler filtrelenerek diğer organizasyonun içine yönlendirilir. Yani erişim yetkisi olmayanlar güvenlik duvarı sayesinde engellenmektedir. Güvenlik duvarları sadece organizasyonlar arasında değil kurumun iç ağıyla internet arasında da tanımlanır.

Güvenlik duvarından geçerek ağın içerisine gelen erişimler STS'ler tarafından izlenmektedir. STS'ler ağ içerisindeki her bileşen üzerinde gerçekleşen eylemleri takip etmektedir. Yani ağ trafiği STS'ler tarafından sürekli gözetim altında tutulmaktadır. STS'ler saldırı olduğunu düşündüğü eylemlere müdahale eder ve ağ yöneticisine bilgi verir.

STS'ler ve güvenlik duvarları birbirlerini bütünleyici iki güvenlik unsuru olarak düşünülebilir. Güvenlik duvarları evinizin önüne koyduğunuz barikat ve STS'ler eve kurulan güvenlik sistemi olarak düşünülebilir.



Şekil 2.3: Güvenlik duvarı.

2.4. MAKİNE ÖĞRENMESİNE GENEL BAKIŞ

2.4.1. Makine Öğrenmesi

Makine öğrenmesi, insanların öğrenme yöntemlerini bilgisayarlara da kazandırabilmek için araştırılan bir alandır. Makine öğrenmesinde geçmişteki veriler incelenerek gelecekle ilgili tahminde bulunulur. Bu kapsamda temel hedef kendi kendine öğrenebilen ve kendi kendini geliştirebilen sistemler oluşturmaktır. Makinelerin öğrenme, tahminleme ve karar verme aşamalarında çeşitli algoritmalar kullanılmaktadır. Her öğrenme süreci iki aşamadan oluşmaktadır. Bunlar;

- Belirli bir küme üzerinde bir sistemdeki bilinmeyen bağımlılıkların tahmini
- Tahmini bağımlılıkları kullanarak sistemin çıktılarının tahmin edilmesi

Makine öğrenmesi algoritmaları yüksek boyutlu veri desenlerinin tanınmasını sağlayan güçlü bir veri eksenli çıkarsama araçları grubudur [15]. Algoritmanın çalışması temel olarak iki veri seti üzerinde gerçekleştirilir. Bunlardan ilki öğrenmenin gerçekleştirildiği eğitim kümesi diğeri ise öğrenilen bilgilerin test edildiği test veri kümesidir. Makine öğrenmesinde öğrenme ikiye ayrılmaktadır. Bunlar;

- Denetimli Öğrenme (Supervised learning)
- Denetimsiz Öğrenme (Unsupervised learning)

2.4.1.1. Denetimli Öğrenme

Denetimli öğrenme, girdi kümesinin belli ve etiketli olduğu, bunun sonucunda bir çıktı üretmeye çalışan öğrenme tipidir. Denetimli öğrenme, etiketli eğitim veri kümesi içerisinde yer alan giriş değerleri ile çıkış değerlerini eşleştirmeyi hedefler.

Denetimli öğrenme bir sınıflandırma problemi olarak düşünülebilir. Sınıflandırma görevi, girdileri sonlu sayıda sınıflar olarak kategorize ederek öğrenmeyi ifade etmektedir. Sınıflandırma algoritmaları denetimli öğrenme yöntemini kullanmaktadır.

Sınıflandırma algoritmalarından bazıları aşağıdaki gibidir.

- Destek Vektör Makinesi (Support Vector Machine - SVM)
- Karar ağaçları

- K-en yakın komşu (KNN - K-nearest neighborhood)
- Naive Bayes

2.4.1.2. Denetimsiz Öğrenme

Etiketsiz veri kümeleri kullanılarak çıkarımlarda bulunmaya çalışan öğrenme tipidir. Çıkarımlar üzerinden ilerlediği için öğrenme sürecinde bir çıktı üretilmemektedir. Denetimli öğrenmeye göre daha uzun sürer. Bunun nedeni etiket bilgisi olmadığı için başlangıç kabul edilen noktanın sonuçtan çok uzak olabilme ihtimalidir.

Denetimsiz öğrenmede amaç genel olarak gruplamaları belirlemektir. Bu sebeple denetimsiz öğrenme, kümeleme problemi olarak düşünülebilir. Kümeleme algoritmalarından en sık kullanılanı;

- K-ortalama (k-means) algoritmasıdır.

Makine öğrenmesinin günlük hayat içerisinde birçok kullanım alanı mevcuttur. Bu uygulamalardan bazıları şunlardır;

- Kredi kartı dolandırıcılığı tespit etme
- Arama motorları
- Konuşma ve El yazısı tanıma
- Hakimler bilgisayar programı - Çin'de pilot uygulama
- IBM - Deep Blue
- IBM - Watson
- Biyoinformatik vb.

2.5. GENETİK ALGORİTMA

Genetik algoritma (GA), biyolojik süreçler içerisinde yer alan doğal seçilim ve genetiğin evrimsel temellerine dayanarak karmaşık problemlere çözüm getirmeyi hedefleyen sistematik bir yöntemdir. GA, 1975 yılında John Holland tarafından optimizasyon problemlerinin çözümü için geliştirilmiştir. GA'lar arama uzayı içerisinde daha iyi performans gösteren çözümleri bularak bunlardan bir küme oluşturmayı

hedeflemektedir. GA'lar çözüm kümesini oluştururken Charles Darwin'in "en iyinin hayatta kalması" ilkesine göre çalışmaktadır.

GA problemin çözümü sırasında oluşturulan bireyler içerisinde en güçlü olanların popülasyonda kalmalarını simüle etmeye çalışmaktadır. Her birey, arama uzayındaki olası bir çözümü ifade etmektedir ve popülasyondaki bireyler daha iyi çözümler üretmek için evrim sürecine dahil edilmektedir.

GA'lar aşağıdaki kuralları baz alarak çalışmaktadır.

- Bir popülasyondaki bireyler kaynaklar için rekabet etmektedir.
- Her rekabetin sonucunda başarılı olanlar yeni bireylerin üretimine zayıf olanlardan daha fazla katkı sağlamaktadır.
- İyi bireylerden gelen genler popülasyon boyunca aktarılarak daha iyi bireyler üretmek için kullanılmaktadır.
- Oluşturulan her nesil gittikçe daha iyi bir çözüm kümesine dönüşecektir.

Popülasyon içerisindeki bireyler belli bir uzunlukta ve belli değerleri alabilecek şekilde kodlanmaktadır. Bireyler kromozom olarak düşünülebilecekken aldıkları her bir değer gen olarak düşünülebilir. Kodlama için kullanılan farklı yöntemler mevcuttur. Bunların hangisinin tercih edileceği probleme özgüdür [16]. Kodlama için kullanılan teknikler aşağıdaki gibidir.

- **İkili Kodlama:** Genler 0 ve 1 değerlerinden oluşmaktadır.
- **Permütasyon Kodlama:** Gezgin satıcı gibi sıralama problemlerinde kullanılan bir yöntemdir. Her kromozom bir dizideki sıralı sayıları ifade etmektedir.
- **Değer Kodlama:** Kromozomlar problemin türüne göre farklı değerlerden oluşabilmektedir. Değerler karakter veya gerçek sayılardan oluşabilmektedir.
- **Ağaç Kodlama:** Genetik programlama için geliştirilen programlar veya ifadeler için kullanılır. Genler fonksiyon veya komut gibi değerleri alabilmektedir.

Bireylerin popülasyon içerisinde kalıp kalmayacaklarına uygunluk değeri ile karar verilmektedir. Çözüm kümesinin içerisine dahil etmek için optimal veya optimale yakın uygunluk değerine sahip bireyler aranmaktadır. Uygunluk değeri probleme özgü bir hesaplama yöntemine bağlı olarak geliştirilmektedir. GA kromozomlardan gelen genleri

belli kurallar çerçevesinde birleştirerek en iyi çözümleri üretmeyi hedeflemektedir. Popülasyon içerisinden ebeveynlerin seçilmesi uygunluk değerine göre yapılmaktadır. Bu ebeveynler belli kurallar çerçevesinde yeni bireyler oluştururlar ve bu işlem en iyi bireylerin elde edilmesine yani çözüm kümesinin oluşturulmasına dek bu şekilde devam etmektedir.

2.5.1. Genetik Algoritma Operatörleri

GA'nın uygulanması aşamasında takip edilen birkaç yöntem mevcuttur. Başlangıç popülasyonu rastgele belirlendikten sonra 3 temel genetik operatör kullanılmaktadır. Bu operatörlerin temel işlevleri aşağıdaki gibi özetlenebilir.

- Uygunluk değeri en uygun olanın seçilmesine dayanan seçme operatörü
- Bireyler arasında değişim işlemini gerçekleştiren çaprazlama operatörü
- Son olarak rastgele değişikliklerin yapıldığı mutasyon operatörü

2.5.1.1. Seçme Operatörü

Darwin'in "en iyinin hayatta kalması" ilkesine dayandırılarak uygunluk değeri iyi olan bireylere öncelik verilmekte ve bu bireyler hiçbir değişikliğe uğratılmadan ebeveyn olarak seçilmektedir. Temel olarak iyi genlerin gelecek kuşaklara aktarılması hedeflenmektedir. Seçim işlemi için kullanılan farklı yöntemler mevcuttur. Bunlardan bazıları aşağıdaki gibidir.

- **Rulet Çarkı Yöntemi:** Süreç, gelecek nesillerin temellerini oluşturmak için popülasyon içerisinden stokastik bir şekilde ebeveyn seçmeye dayanır. Uygunluk değeri yüksek olan bireylerin seçilme şansları zayıf bireylere göre daha yüksektir. Zayıf bireylerin tamamiyle seçilme şansı yoktur denilemez. Doğada olduğu gibi bu bireylerin de gelecek nesillere aktarabilecekleri yararlı genetik özellikleri olabilir. Popülasyon içerisindeki bireylerin uygunluk değerlerinin toplamı bulunur. Rastgele olarak bir x değeri belirlenir. 0'dan başlayarak popülasyon değerleri tekrar toplanır ve x değerinden büyük olunmasını sağlayan birey ebeveyn olarak seçilir.

- **Turnuva Yöntemi:** Popülasyon içerisinden rastgele bireyler seçilerek kendi aralarında bir turnuvaya girmeleri sağlanır. Uygunluk değerlerine göre galip gelen bireyler ebeveyn olmak için seçilmiş olur.

- **Sıralama Yöntemi:** Bireyler öncelikle uygunluk değerlerine göre sıralanır ve daha sonra bunlara birer sıra verilir. Uygunluk değeri yüksek olan birey en yüksek sırayı alırken, en düşük uygunluğa sahip birey en düşük sırayı almaktadır [17]. Her bireye sırasına göre bir seçilme olasılığı verilir ve seçim işlemi bu olasılık değerlerine göre yapılır [18].

2.5.1.2. Çaprazlama Operatörü

GA'yı diğer optimizasyon tekniklerinden ayıran en önemli özelliktir. Seçim operatörü kullanılarak popülasyon içerisinden iki birey seçilir. Belirlenen çaprazlama oranı kapsamında çaprazlanma işlemi gerçekleştirilir. Çaprazlama yöntemlerine aşağıdaki metotlar örnek verilebilir [19].

- **Tek Noktalı Çaprazlama:** Rastgele bir k noktası seçilip bu noktadan sonrası ebeveynler arasında çaprazlanarak yeni bireyler oluşturulmaktadır.
  k = 4 olmak üzere ;

  1.Ebevyn Kromozom : 1 0 1 0 | 1 0 0 1 0

  2.Ebevyn Kromozom : 1 0 1 1 | 1 0 1 1 0

  1.Çocuk Kromozom : 1 0 1 0 | 1 0 1 1 0

  2.Çocuk Kromozom : 1 0 1 1 | 1 0 0 1 0

- **İki Noktalı Çaprazlama:** Rastgele 2 nokta seçilerek bu noktalar arasındaki genlerin çaprazlanması ile yeni bireyler oluşturulmaktadır.
  k = 3 ve k = 7 olmak üzere ;

  1.Ebevyn Kromozom : 1 0 1 | 0 1 0 0 | 1 0

  2.Ebevyn Kromozom : 1 0 1 | 1 1 0 1 | 1 0

  1.Çocuk Kromozom : 1 0 1 | 1 1 0 1 | 1 0

  2.Çocuk Kromozom : 1 0 1 | 0 1 0 0 | 1 0

2.5.1.3. Mutasyon Operatörü

Biyolojik mutasyon baz alınarak bir nesilden diğer nesile genetik çeşitliliğin aktarılması için kullanılan operatördür. Basitçe ifade etmek gerekirse mutasyon yeni çözüm elde etmek için kromozomda rastgele küçük değişimler yapma işlemidir.

Her bireyin mutasyona uğraması gerekmemektedir. Mutasyon operatörünün kullanılması bir olasılık değerine bağlıdır. Mutasyon operatörü çözüm kümesi oluşturulmaya çalışılırken yerel optimum noktalarda çözümün sıkışmasının önüne geçmeyi hedeflemektedir. Mutasyon yöntemlerine aşağıdaki metotlar örnek verilebilir [20].

- **Değer Değiştirme Yöntemi:** Kromozoma ait genlerden birinin 1 iken 0 ya da tam tersinin yapılması ile gerçekleştirilmesi işlemidir.
- **Tersine Çevirme Yöntemi:** Kromozom içerisinden bir aralık seçilerek bu genlerin kendi aralarında dizilimlerinin ters çevrilmesi ile gerçekleştirilen yöntemdir.
- **Ekleme Yöntemi:** Kromozom içerisinden bir gen seçilerek bunun kromozom içerisinde başka iki genin arasına eklenmesi ile gerçekleştirilir.
- **Karşılıklı Değişim Yöntemi:** Kromozom içerisinden rastgele iki nokta seçilerek bu noktalardaki genlerin karşılıklı olarak değiştirilmesi sağlanmaktadır.

2.5.2. Genetik Algoritma ile ilgili Çıkarımlar

- Uygunluk fonksiyonu problem ve genetik kodlama bazında belirlenmesi gereken bir hesaplama yöntemidir.
- Seçim operatörünü tek başına kullanmak popülasyonda sürekli en iyi bireylerin yer almasına neden olacaktır.
- Seçim ve çaprazlama operatörlerinin kullanılması, algoritmaların iyi fakat optimal olmayan çözümler üretmesine neden olacaktır.
- Mutasyon operatörünün tek başına kullanılması rastgelelik oranını arttırıp çözüm kümesini optimal çözümlerden uzaklaştıracaktır.
- Arama uzayının büyümesi çözüme ulaşılacak süre ile doğru orantılıdır. Arama uzayı arttıkça en iyi çözümlere ulaşma süresi de artmaktadır.

2.6. BAĞIŞIKLIK SİSTEMİ

2.6.1. İnsan Bağışıklık Sistemi

Bağışıklık sistemi, insan fizyolojisinde hastalıklara karşı savaşan karmaşık bir mekanizmadır. Vücuda giren veya temas edilen her hücrenin analizini yaparak sağlıklı hücreler olup olmadığına karar verir ve sağlıklı olmayan hücrelerle savaşarak vücudu hastalığa karşı korumaya çalışır. Bağışıklık sisteminin ilk görevi bu hücrelerin vücuda girmesini engellemek iken, vücuda girmeleri halinde bunları vücuda girdikleri yerde yok etmek, yayılımlarını engellemek ve sonrasında yayılımı geciktirmek gibi çeşitli görevleri mevcuttur. Bu mekanizma içerisinde öz ve öz olmayan hücrelerin ayrımı yapılarak patojen ve tümörler yok edilir ve vücuda zarar verilmesi engellenmeye çalışılır. Bağışıklık sisteminin zararlı maddelere yeterli düzeyde bağışıklık yanıtı üretememesinin yanısıra öz ve öz olmayan hücrelerin ayrımını yapamadığı durumlarda meydana gelen farklı hastalıklar da mevcuttur. Bağışıklık sistemi içerisinde görevli yapı ve hücreler arasında kuvvetli bir etkileşim ve iletişim ağı mevcuttur. Bağışıklık sistemi bileşenleri bu yapı sayesinde birbirlerini tamamlayıcı bir şekilde çalışmaktadır.

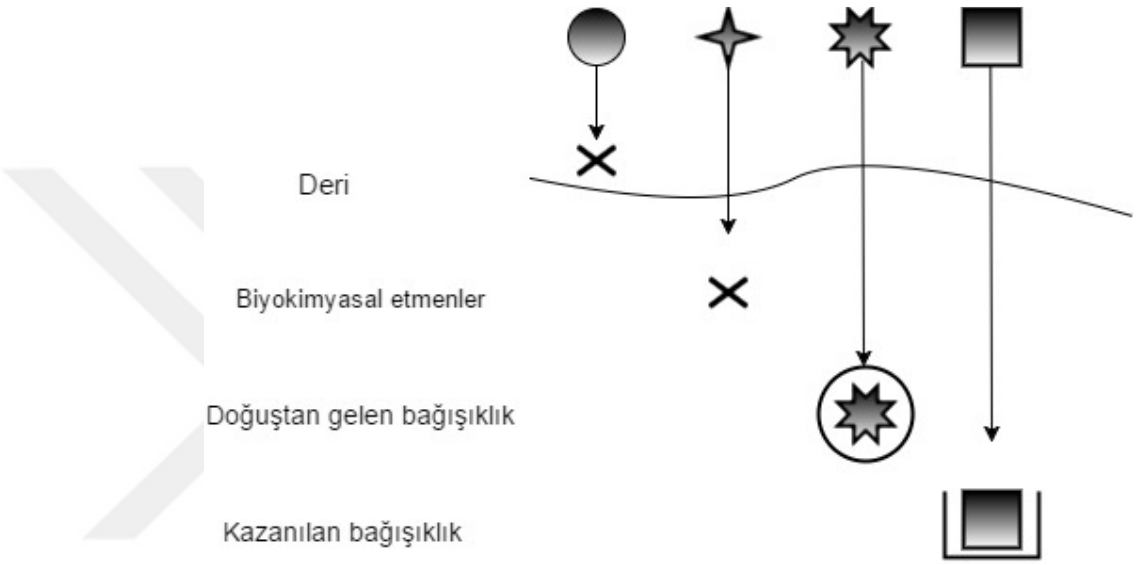
Bağışıklık sisteminin çok katmanlı bir koruma yapısı mevcuttur. Bu katmanlar sırasıyla aşağıdaki gibidir.

- **Deri:** Bakteri ve virüslere karşı ilk savunmayı yaparak vücuda girmelerini engellemeye çalışır.

- **Biyokimyasal etmenler:** Midemizde yer alan asit, yiyecek ve içeceklerden alınan bakterileri parçalar. Ter ve yağ bezlerinin üstlendikleri görevde bu grupta kategorize edilebilir.

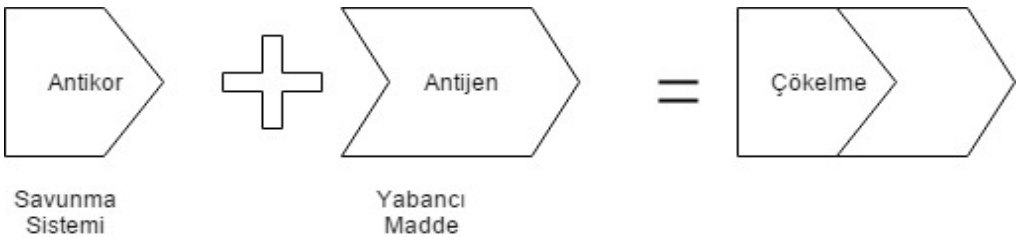
- **Doğuştan gelen bağışıklık:** Doğal bağışıklık sistemimiz doğduğumuz andan itibaren oluşan ve öz olmayan hücrelerin tespiti için birçok hücre, molekül barındıran bir sistemdir [21]. Doğuştan gelen bağışıklık sistemi soysal olarak tanıdığı virüslere karşı hemen tepki verir ancak koruması uzun süreli değildir. Virüsleri tanıması kazanılmış bir davranış olmadığı için farklı türdeki patojenlere karşı tepki veremeyebilir.

- **Kazanılmış bağışıklık:** Doğal bağışıklık sisteminden farklı olarak tanımadığı maddeler ile karşılaştığında bunları yok etmeye çalışır. Bu sistemin temel özelliği öğrenmeye dayanmaktadır. Öğrendiği bilgiyi hafıza hücrelerinde tutarak bir sonraki karşılaşmada hangi tür hücrelerle savunma gerçekleştireceğini bilir ve daha hızlı aksiyon alır.



Şekil 2.4: Doğal bağışıklık sistemi.

Kazanılmış bağışıklık sistemi içerisinde savunma için kullanılan bileşenlerin en önemlilerinden biri lenfositlerdir. Kan ve lenf sistemi içerisinde milyonlarca lenfosit mevcuttur. Savunma bunların ürettikleri antikorlar aracılığıyla yapılır. Vücuda giren yabancı maddeler için bağışıklık sistemi aşağıdaki gibi çalışır.



Şekil 2.5: Antikor ve Antijen reaksiyonu.

Lenfositler üçe ayrılmaktadır. Bunlar;

- **Doğal öldürücü hücreler:** Doğal bağışıklık sisteminin bir üyesidir. Tümör ve virüsler tarafından enfekte olmuş hücrelere saldırırarak bunları yok etmeyi hedeflemektedir.

- **B lenfosit:** Kemik iliğinde olgunlaşarak bellek ve plazma hücreleri oluşturur. Bu lenfosit türü yardımcı T hücresi tarafından uyarıldıktan sonra antikor üretmeye başlar. Bellek hücreleri karşılaştıkları antijenlere özel olup uzun süre yaşayabilmektedir. Böylece aynı antijenle karşılaşılması durumunda hızlı cevap üretebilmektedir. Plazma hücreleri ise antikor üreterek antijenlere bağlanır ve antijenlerin yıkımı için diğer bağışıklık sistemi bileşenlerine yardımcı olur.

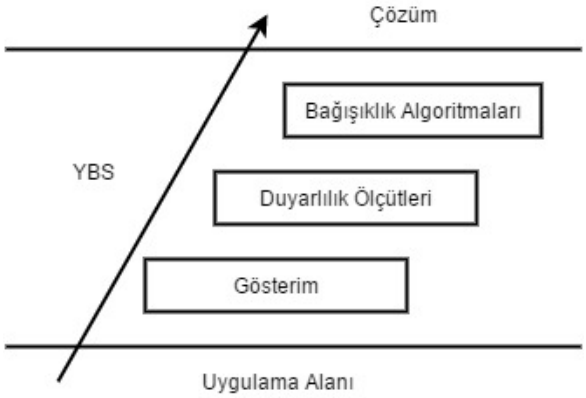
- **T lenfosit:** Bağışıklık sisteminin önemli üyelerinden olan timüs bezinde olgunlaşır. Hem hücresel bağışıklıktan hem de B lenfositlerin aktivasyonundan sorumludur. Virüs taşıyan hücreleri belirleyerek bu hücreleri öldürmektedir.

2.6.2. Yapay Bağışıklık Sistemi

YBS insan bağışıklık sisteminin teorileri ve modellerinden esinlenerek oluşturulmuş ve karmaşık problemlerin çözümü için uygulanan bir sistemdir [22]. Bilgi işleme açısından bakıldığı zaman YBS, öğrenme, bellek ve ilişkisel erişim yeteneklerine sahip paralel ve dağıtık bir sistemdir [23].

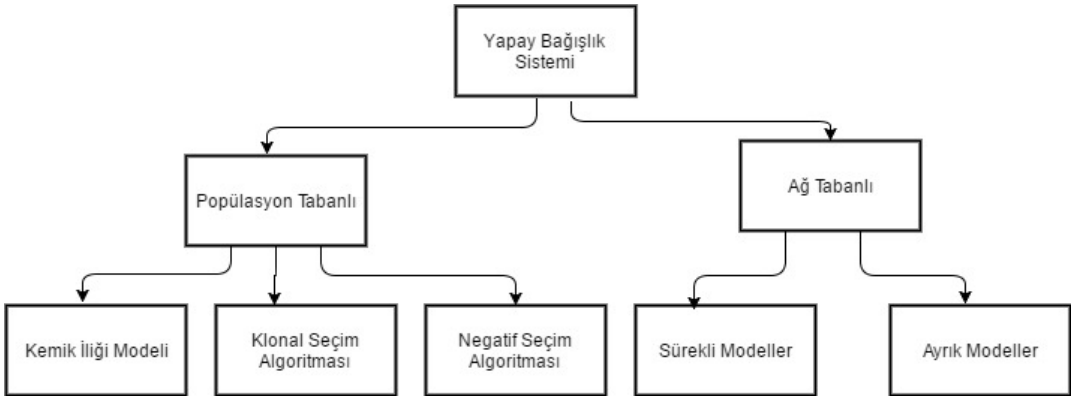
YBS, örüntü tanıma, optimizasyon, anormallik tespiti, hata teşhisi, robotik, veri analizleri gibi karmaşık hesaplama veya mühendislik problemlerini çözmeyi amaçlamaktadır.

YBS çok katmanlı bir yapıya sahiptir. Bu yapı öncelikle problemin gösterim şeklinin belirlenmesi sonrasında gösterime bağlı olarak duyarlılık ölçütlerinin belirlenmesi ve son olarak da ilgili bağışıklık algoritmasının seçimini kapsayan bir süreçtir [24]. YBS'nin katmanlı yapısı Şekil 2.6'da gösterilmiştir [25].



Şekil 2.6: YBS'nin çok katmanlı yapısı [25].

Literatürde farklı problemlerin çözümü için kullanılan farklı YBS algoritmaları mevcuttur. Bu algoritmaları Şekil 2.7'de olduğu gibi gruplandırmak mümkündür. YBS algoritmaları iki gruba ayrılmaktadır. Bunlardan ilki popülasyon tabanlı, ikincisi ise ağ tabanlı algoritmalardır.



Şekil 2.7: Yapay Bağışıklık Sistemi algoritmaları [26].

Bu çalışma kapsamında popülasyon tabanlı algoritmalardan olan negatif seçim algoritması kullanılmıştır. Bu bölüm kapsamında YBS algoritmaları içerisinde en sık kullanılan iki yöntem olan negatif seçim algoritması ve klonal seçim algoritması ile ilgili teorik bilgilere yer verilmiştir.

2.6.2.1. Negatif Seçim Algoritması

Negatif seçim algoritması, diğer YBS algoritmalarında olduğu gibi doğal bağışıklık sisteminin modellenmesi sonucu oluşturulmuştur. NSA, timusta gerçekleşen T hücresi olgunlaşması süreci üzerinden tasarlanmıştır. Timusta yer alan T hücreleri kendi hücreleri tarafından tanınmazsa bağışıklık sistemine katılmadan elenir [27].

Negatif seçim algoritmasının çıkış noktası bilgisayar virüsleri olmuştur. Gelişen ağ yapıları ve artan kullanıcı sayısı ile birlikte bilgisayarlarımız için tehdit oluşturan virüsler de artış göstermiştir. Bu virüslerin tespit edilmesi ve sonrasında önlem alınması ile ilgili birçok çalışma yapılmış, ancak bu gelişmelere rağmen virüslerin bilgisayarlarımızdan tamamen atılması maalesef mümkün olmamıştır. Bu noktada YBS ile üretilen çözümler son yıllarda oldukça etkili olmuştur [28].

Bağışıklık ile ilgili ilkelerin bilgisayar güvenliğinde kullanılması 1994 yılında Stephanie Forrest ve ekibi tarafından New Mexico üniversitesinde yapılan bir çalışmayla başlamıştır [29]. Bu çalışma kapsamında geliştirilen bağışıklık tabanlı sistem mevcut işletim sistemlerinden daha karmaşık kavramlara sahip ve aynı zamanda bunların sağladığından daha fazla koruma imkanı vermektedir [30].

Bilgisayar sistemlerinin güvenliği genel olarak yetkisiz erişimin engellenmesi, veri bütünlüğünün korunması ve virüs yayılımlarının engellenmesi gibi konuları kapsamaktadır [29]. Bilgisayar sistemlerini zararlı virüslerden koruma sorunu kendi bileşenlerini yabancı bileşenlerden ayırma sorununun bir örneği olarak görülmektedir. Negatif seçim algoritması olarak adlandırılan bu yöntem bilgisayar güvenliğine yönelik kriptografik ve deterministik yaklaşımları tamamlayıcı niteliktedir [30]. Negatif seçim algoritması ilk olarak virüs tespitinde dosya doğrulama yöntemi olarak kullanılmıştır. Bu yöntem sistem içerisinde uygun olmayan bir durumun meydana gelmesi sonucunda kullanılmaktadır.

Negatif seçim algoritmasında ilk aşama olarak öz hücreler tanınmaya çalışılır. Daha sonra rastgele bir şekilde detektörler üretilmeye başlanır. Bu detektörlerden öz hücreler ile eşleşmeyenler detektör kümesine aktarılır. Nümerik verilerle üretilen dedektör ve öz hücreler arasındaki benzerliğin ölçülmesi için Öklid, Manhattan ve Minkowski gibi mesafe ölçümleri kullanılırken [31], string değerlere sahip vektörler arasındaki mesafe

ölçümü için r-contiguous ve r-chunk gibi metotlar kullanılmaktadır [32]. Negatif seçim algoritması virüs tespitlerinde, bilgisayar ve ağ saldırıları, zaman serileri tahmini gibi problemlerde yaygın olarak kullanılan bir algoritmadır.

Aşağıdaki eşitliklerde D kısaltması mesafeyi, Ab antikoru, Ag antijeni, Ɛ eşik değerini, n ise özellik sayısını ifade etmektedir.

Öklid:

$$D = \sqrt{\sum_{i=1}^{n} |Ab_i - Ag_i|^2} \quad (2.1)$$

Manhattan:

$$D = \sum_{i=1}^{n} |Ab_i - Ag_i| \quad (2.2)$$

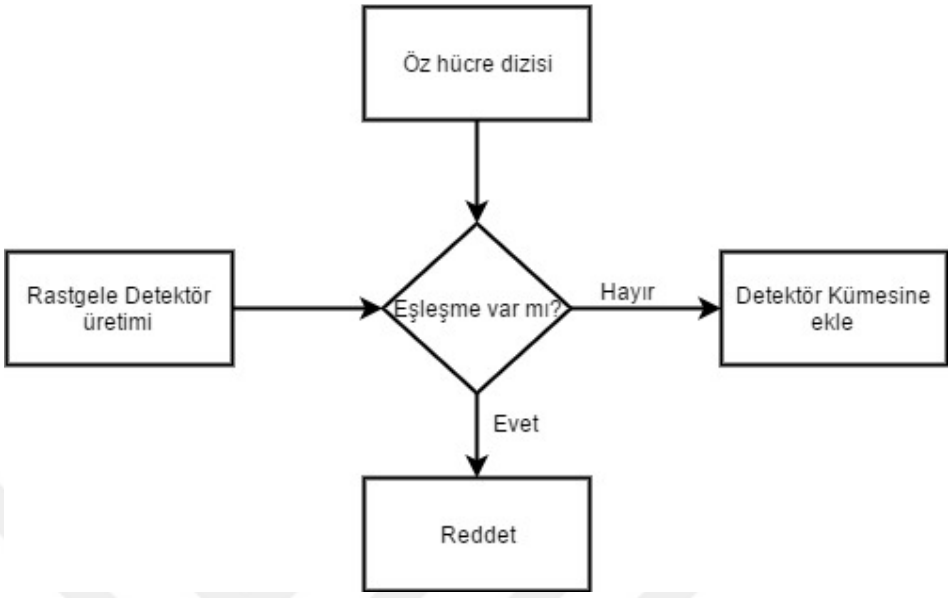
Minkowski:

$$D = \sqrt[\lambda]{\sum_{i=1}^{n} |Ab_i - Ag_i|^{\lambda}} \quad ; \lambda \geq 1 \quad (2.3)$$

$$E = D - \varepsilon \quad (2.4)$$

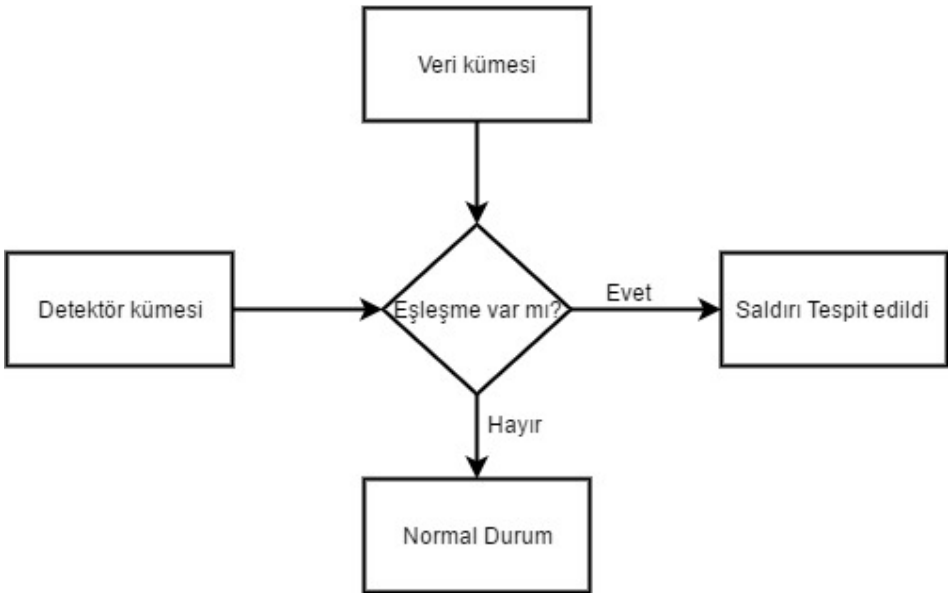
E > 0 ise Ag =[Ag_1, Ag_2,.....Ag_n] olarak ifade edilen detektörler öz hücreler ile eşleşmez ise detektörler kümesine eklenir [33]. Bu detektörler daha sonra test aşamasında kullanılır. Eğer yeni veri kümesinde bunlarla eşleşen bir veri olursa beklenmeyen durum olarak işaretlenir.

Negatif seçim algoritması iki aşamadan oluşmaktadır. Şekil 2.8'de negatif seçim algoritmasının ilk aşaması olan detektör üretimi gösterilmiştir. Bu aşamada üretilen detektörler ikinci aşama olan saldırı tespiti kısmında kullanılmaktadır. Şekil 2.9'da ise negatif seçim algoritmasının ikinci aşaması olan saldırı tespiti gösterilmiştir.



Şekil 2.8: Negatif Seçim Algoritması detektör üretim aşaması.

Aday detektör ve öz hücreler arasındaki benzerliğin ölçülmesi için bu çalışma kapsamında Öklid mesafe ölçümü kullanılmıştır. Aynı şekilde saldırı tespit aşamasında da benzerliğin ölçümü için bu yöntem tercih edilmiştir.



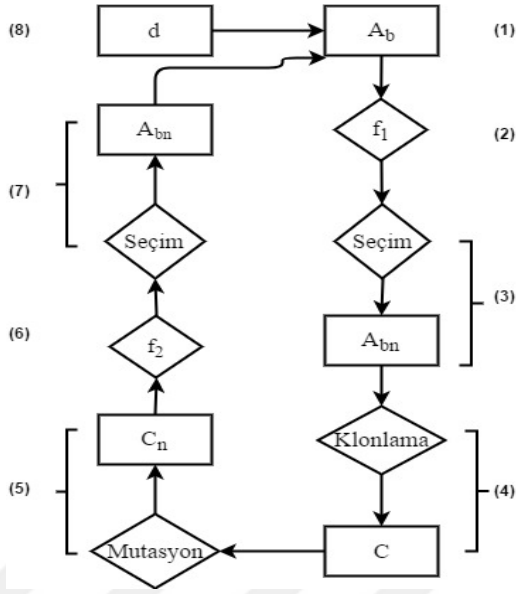
Şekil 2.9: Negatif Seçim Algoritması saldırı tespit aşaması.

2.6.2.2. Klonal Seçim Algoritması

Klonal seçim algoritması 1959 yılında Burnet tarafından geliştirilmiştir [34]. Antijenlerin B lenfositleri tarafından tanınmasından sonra verilen bağışıklık uyarısını açıklamak için kullanılan algoritmadır [25]. Sadece antijenleri tanıyan hücrelerin çoğalması için seçildikleri fikrini ortaya koyar [35]. Bu algoritmaya göre her yapılan işlem sonucunda ulaşılmak istenen noktaya yaklaşılmalıdır [28]. Biçim tanıma ve optimizasyon gibi karmaşık süreçleri olan problemler için kullanılır.

Klonal seçim algoritması iki temel konu üzerinde modellenmiştir. Bunlardan ilki sadece antijeni tanıyan hücrelerin çoğalma için seçilmesi, ikincisi ise seçilen ve çoğalan hücrelerin duyarlılık olgunlaşması işlemine tabi tutularak antijene olan duyarlılıklarının arttırılmasıdır [36].

Doğal bağışıklık sistemimiz içerisinde yer alan hafıza hücrelerine sahiptir. Yani en çok kullanılan antikorlar hafızada tutularak bunlar klonlanmakta, kullanılmayanlar ise bir süre sonra yok edilmektedir. Benzerliğin tam olarak sağlanması için olgunlaştırma ve yeniden seçme gibi süreçler desteklenmiştir. Klonal seçim Darwin'in evrim teorisinde yer alan üç temel fonksiyonu kullanmaktadır. Bu fonksiyonlar sırasıyla farklılaştırma, çeşitlendirme ve doğal seçilim mekanizmalarıdır [37]. Klonal seçim algoritmasının akış diyagramı ve adımları aşağıdaki gibidir [28].



Şekil 2.10: Klonal Seçim Algoritması [28].

1. **Adım:** Antikorlar başlangıç çözüm kümesi olarak kabul edilir (A_b).
2. **Adım:** Popülasyondaki tüm antikorlar için benzerlik dereceleri hesaplanır (f_1).
3. **Adım:** n adet en yüksek benzerlik derecesine sahip antikor seçilir (A_{bn}).
4. **Adım:** Seçilen n adet antikorun benzerlik dereceleriyle doğru orantılı olacak şekilde antikorlar klonlanır. Yani benzerlik ne kadar yüksekse ilgili antikordan üretilen klonlar da o orantıda fazla olacaktır (C).
5. **Adım:** Antikorlar benzerlik dereceleriyle ters olacak şekilde mutasyona uğratılır (C_n).
6. **Adım:** Mutasyona uğratılmış antikorlar için benzerlik oranı tekrar hesaplanır (f_2).
7. **Adım:** n adet antikor yüksek benzerlik derecesine sahip olacak şekilde yeniden seçilir (A_{bn}).
8. **Adım:** d adet en düşük dereceli antikorlar seçilerek yeniden üretilen antikorlar ile yer değiştirilir.

2.7. ÖZELLİK SEÇİMİ VE BOYUT İNDİRGEME

2.7.1. Özellik Seçimi

Özellik seçimi, veri kümelerindeki özellik sayılarının artmasıyla birlikte makine öğrenmesi için önem kazanmış bir konudur. Birbiriyle bağlantılı verilere karar vererek bunlar üzerinde işlem yapmak hem performans açısından hem de üretilen sonuçların verimliliği açısından avantaj sağlamaktadır. Özellik seçimi, veri kümesi içerisinde birbiriyle bağlantılı olmayan, alakasız ve fazla özelliklerin çıkartılması sonucu doğruluk oranı yüksek olan modeller oluşturmayı hedeflemektedir. Özellik alt kümesi oluşturmak için kullanılan farklı yöntemler mevcuttur. Bunlar;

- **Filtreleme tabanlı**: İstatistiksel bir yöntemdir. Her özelliğe bir puan atayarak onun veri kümesinde kalıp kalmayacağına karar vermeye çalışır. Bilgi kazancı yöntemi en sık kullanılan filtreleme tabanlı algoritmalardan biridir. Bilgi kazancı veri kümesindeki her özellik için hesaplanır. Bu değerin hesaplanabilmesi için özelliklerin entropi'si (özelliğin bilgi içeriği) bulunur. Entropi veri kümesi içindeki belirsizlik seviyesinin ölçümünü sağlar ve entropi değeri yüksek olan özelliklerin veriyle ilgili daha fazla bilgiye sahip olduğu kabul edilir [38]. Entropi hesaplama formülü ve detayları aşağıdaki gibidir [38].

$$E(V) = -\sum_{i=1}^{m} p_i \log_2 p_i \quad (2.5)$$

Yukarıdaki formülde yer alan p_i değeri, V veri kümesi içerisindeki i sınıfının olasılığı olarak ifade edilmektedir. Olasılık değeri i sınıfına ait örnek sayısının toplam örnek sayısına bölünmesi ile elde edilir. E(V) ise veri kümesinin entropi değeri olarak ifade edilir.

Entropi hesaplaması sonrasında her özellik için bilgi kazanımı hesaplanır ve yüksek kazanıma sahip olan özellikler seçilerek bunların veri kümesi içerisinde kalması sağlanır. Formül ve detaylar aşağıdaki gibidir [38].

$$G(V,X) = E(V) - \sum_{i=1}^{n} p(V_i)E(V_i) \quad (2.6)$$

G(V,X), V veri kümesinin n tane alt bölüme X özelliğinden bölünmesi sonucunda elde edilecek bilgi kazancını ifade etmektedir. $E(V_i)$, i alt bölümünün

bölünmeden sonraki entropi değerini, $p(V_i)$ ise i alt bölümünün bölünmeden sonraki olasılığını ifade etmektedir.

- **Sarmal tabanlı:** Özellikler içerisinden farklı kombinasyonların hazırlanıp değerlendirilerek ve sonrasında diğer kombinasyonlarla karşılaştırılıp bir küme seçmeyi hedefleyen arama problemi olarak düşünülebilir. Farklı özelliklerin kombinasyonunu değerlendirip modelin doğruluğuna göre puan atayan tahmine dayalı bir model kullanılır [39].

- **Gömülü:** Öğrenme modeli oluşturulurken hangi özelliklerin modelin doğruluğuna katkıda bulunduğunu öğrenmektedir ve genellikle geliştirildikleri öğrenme makinalarına özgüdür [40].

Filtreleme tabanlı yaklaşımlar sınıflandırıcılara bağımlı değildir ve genellikle sarmal tabanlı yöntemlerden daha hızlı ve daha ölçeklenebilirdir [41]. Ek olarak sarmal tabanlı yöntemler ile karşılaştırıldıklarında daha düşük hesaplama karmaşıklığına sahip oldukları görülmektedir [41].

2.7.2. Boyut İndirgeme

Veri kümelerindeki boyutların artışıyla birlikte uygulamaların performanslarında düşüş meydana gelebilmektedir. Bu kapsamda boyut indirgeme yöntemleriyle veri kümeleri içerisindeki etkili özellikleri ayıklayarak modeli basitleştirmek, sınıflandırma aşamasında hesaplama maliyetlerinin düşürülmesi ve daha doğru sonuçların elde edilmesi için etkili bir yöntemdir [42].

Boyut indirgeme ile veri kümesi içerisinden farklı altkümeler seçerek yeni bir küme oluşturulur. Gerçekleşme maliyeti, depolama ve ölçümleme parametreleri açısından etkili olmaktadır. Makine öğrenmesi uygun veri tasarımı ile başlar ve daha iyi bir performans sergileyebilmesi için veri kümesinden türetilmiş yeni veri kümesi ile daha iyi sonuçlar elde edilebilir. Boyut indirgeme ile ilgili iki önemli hedef bulunmaktadır. Bunlar;

- Verilerin en iyi şekilde yeniden yapılandırılmasını sağlamak
- En verimli tahmini yapabilmek

Boyut indirgeme işlemini gerçekleştirebilmek için farklı yöntemler mevcuttur. Bunlara örnek olarak aşağıdaki yöntemler gösterilebilir.

- Kümeleme
- Temel doğrusal dönüşümler (Temel Bileşenler Analizi, Doğrusal Diskriminant Analizi)
- Spektral dönüşüm gibi daha gelişmiş doğrusal dönüşümler (Fourier)

2.7.2.1. Temel Bileşen Analizi (Principal Component Analysis - PCA)

TBA 1901 yılında Karl Pearson tarafından bulunmuştur. Ancak daha sonra 1933 yılında Harold Hotelling tarafından bağımsız olarak geliştirilmiştir. Aynı zamanda temel bileşen kavramı literatüre Hotelling tarafından kazandırılmıştır [43].

TBA başlangıç kümesindeki özelliklerin lineer bir kombinasyonunu üreten özellik çıkarım tekniğidir [44]. TBA veri içerisindeki korelasyon değerlerinin belirlenmesiyle ilgilidir. Aralarında korelasyon bulunan yani birbiriyle ilişkili verilerden, veriler arasındaki doğrusallık sağlanarak aralarında korelasyon bulunmayan özelliklerin elde edilmesi hedeflenmektedir. Yeniden oluşturulan bu özelliklere temel bileşen (TB) denilmektedir.

TB'ler, veri kümesi içerisindeki özelliklerin doğrusal olarak karşılıklarını ifade etmektedir. Bunların sayıları orijinal özellik sayısına eşit veya daha az olabilir [44]. Özellik sayısı azaldığı için bir sonraki aşamalarda gerçekleştirilecek hesaplama maliyetlerinin düşürülmesinde kazanç sağlamaktadır. TB bulma işlemi sırasında yapılan dönüşüm ilk TB'nin en büyük varyansa sahip olacağı şekilde yapılmaktadır. Bir sonraki TB, ilk TB'ye dikey olacak şekilde seçilmektedir. İlk TB'ye göre daha düşük varyansa sahip olmaktadır ve bu notasyon son TB'ye kadar aynı şekilde ilerlemektedir.

TBA, verideki parçaların belirlenmesi ve bu parçaların benzerliklerini ve farklarını (istatistiksel özellikleri) vurgulayarak verinin vektör olarak gösterimini sağlamaktadır. Birbirleriyle ilişkili verilerin en fazla değişim gösterdiği boyutlar bulunur. Bu

değişimler mümkün olduğunca korunarak boyut indirgeme işlemi gerçekleştirilir [45]. Asıl amaç varyansı yüksek tutarak minimum derecede kayıpla çalışma yapabilmektir [44].

TBA, N boyutlu bir veri kümesi için N × N boyutunda kovaryans matrisi oluşturur. Bu matris için öz değer ve öz vektörler bulunur. Bulunan öz vektörler, öz değerlerine göre sıralanmaktadır. Öz değerleri en yüksek P tane öz vektör seçilerek veriler P öz vektöre izdüşürülür. Böylece N boyuttan P boyutuna indirgenmiş bir veri kümesi elde edilir [46].

N – orijinal veri kümesinin boyutu

P – TB sayısı

N > P için her P önceki bileşenlerden hariç tutularak hesaplandığı varyansa yönlendirilir.

Veri kümesi üzerinde TBA uygulandıktan sonra üretilen TB'ler aşağıdaki gibi ifade edilmektedir.

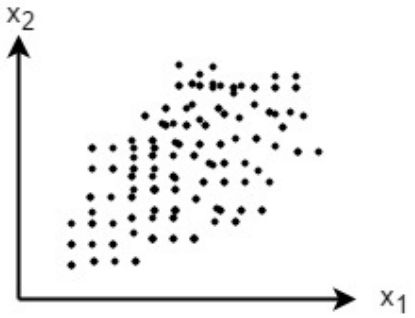
$$TB_1 = a_1X_1 + a_2X_2 + a_3X_3 + a_4X_4 + \ldots\ldots\ldots\ldots + a_NX_N \quad (2.7)$$

TB_1: 1. Temel Bileşen

a_N : X_N için belirlenen katsayı

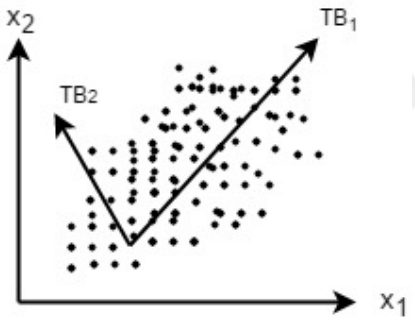
X_N: Orijinal özellik

Aşağıda grafiksel olarak ifade edilen veri için TBA uygulandığında işleme en çok değişim gösterilen eksenden başlanır.



Şekil 2.11: 2 Boyutlu veri kümesinin dağılımı.

TBA, etiket bilgisini kullanmadan analiz yapan denetimsiz bir yöntemdir [47]. Hedef kriter varyansı maksimum tutarak işlem yapmaktır.



Şekil 2.12: Temel bileşenler.

TBA, boyut indirgeme teknikleri arasında en çok kullanılan yöntemlerdendir. Kullanımın yüksek olması aşağıdaki 3 maddeyle özetlenebilir [48].

- Yüksek boyutlu dataları daha düşük boyutlu vektörlere sıkıştıran ve orijinal veri kümesini yeniden oluşturmak için kullanılan optimal lineer bir tekniktir.
- Parametreler doğrudan veri üzerinden örnek kovaryans matrisinin köşegenleştirilmesi yoluyla hesaplanmaktadır.
- Son olarak, verilen parametreler ile sıkıştırma ve açma işlemlerinin gerçeklenmesi kolaylaşır. Çünkü sadece matris çarpımlarına ihtiyaç duymaktadır.

Yüksek boyuttaki verilerin incelenmesi oldukça zor olduğunda TBA'nın farklı alanlardaki kullanımı yaygınlaşmıştır. Kullanım alanları;

- Gen haritalarının incelenmesi (Biyoloji)
- Finansal tahmin üretme (Bankacılık)
- Resim sıkıştırma, görüntü işleme (Bilgisayar bilimleri) vb. alanlardır.

2.7.2.2. Doğrusal Diskriminant Analiz

Doğrusal Diskriminant Analizi (DDA), 1936 yılında Fischer tarafından iki veya daha fazla nesne veya olay sınıfını karakterize eden özelliklerin doğrusal bir bileşimini bulmak için istatistik ve makine öğrenmesinde kullanılmak üzere geliştirilmiş bir yöntemdir [49].

DDA, sınıflar arasında en iyi ayrımı sağlayan vektörleri bularak boyut indirgeme yapmayı amaçlamaktadır [50]. DDA, sınıf içi mesafeyi en aza indirgeyip sınıflar arası mesafeyi maksimize ederek optimal bir dönüşüm çözümü üretmeyi hedeflemektedir [51]. Örüntü tanıma ve görüntü işleme gibi yüksek boyutlu verileri içeren birçok uygulamada yaygın şekilde kullanılmaktadır.

TBA ve DDA arasındaki temel fark TBA'nın özellik sınıflandırması, DDA'nın ise veri sınıflandırması yapmasıdır [52]. TBA orijinal veri kümesinin yapısını değiştirirken DDA yapıyı değiştirmeden daha fazla sınıf ayrımını sağlamaya çalışarak sınıflar arasında karar bölgesi oluşturmaya çalışmaktadır [52]. Bu yöntem aynı zamanda özellik verisinin dağılımını daha iyi anlamaya yardımcı olmaktadır. Optimal dönüşüm, dağılım matrisleri üzerinde öz ayrışmayı uygulayarak kolaylıkla hesaplanabilir. DDA'nın, veri sınıflandırması ve boyut indirgeme olarak iki farklı fonksiyonu vardır.

DDA, lineer olarak ayrılabilen gruplar için kullanılır. Eğer sadece 2 özellik varsa grubu ayırmak için çizgi kullanılır, 3 özellik varsa ayrım için düzlem, eğer daha fazla özellik varsa ayrım için hiper düzlem kullanılmaktadır.

Özetle DDA çoklu sınıf kavramı olan veri kümeleri içerisinde, sınıflar arasındaki ayrımı maksimize eden eksenlerin yönlerini hesaplamaya dayanır.

DDA aşağıdaki 5 adımla özetlenebilir [53].

- Veri kümesi içerisindeki her sınıf için y boyutlu ortalama vektörler hesaplanır.
- Sınıf içi ve sınıflar arası dağılım matrisleri hesaplanır.
- Dağılım matrisleri üzerinden öz değer ve öz vektörler hesaplanır.
- Azalan öz değer değerlerine göre öz vektörler sıralanır ve en büyük öz değerlere sahip n tane öz vektör seçilerek y × n boyutunda yeni bir W matrisi hazırlanır. W matrisinin her kolonu bir öz değere denk gelmektedir.
- y × n boyutlarında oluşturulan yeni matris kullanılarak dönüştürme işlemi tamamlanır. Bu dönüşüm işlemi matematiksel olarak $d = W^T \times k$ şeklinde ifade edilebilir. k, y × 1 boyutunda bir örneği temsil eden vektör iken d, n × 1 boyutunda dönüştürülmüş örneği ifade etmektedir.

3. MALZEME VE YÖNTEM

3.1. KDD CUP 99 VERİ SETİ

STS'ler için tasarım aşaması kadar test edilme aşaması da büyük önem arz etmektedir. Sistemin test edilebilmesi için güvenilir ve kapsamlı veri setlerine ihtiyaç vardır. İnternet üzerinden elde edilen veri setlerinde kayıt bazında detaylı bilgi verilmemektedir. Bu sebeple ağ üzerinde kaydedilen bilginin saldırı olup olmadığı belirtilmediği için öğrenme ve test aşamalarında net sonuçlar üretilememektedir. Gerçek bir ağın veri seti oluşturulmak istendiğinde trafiğin sürekli biri tarafından izlenip kaydedilmesi gerekmektedir. Bu yöntem çok maliyetli olduğu için tercih edilmemektedir. Aynı zamanda bu veri kümelerinin içerisinde farklı saldırı türlerinin olma olasılığı düşük olduğundan çeşitlilik sağlanamamış olacaktır. Bu sebeple bu tarz işlemler için kullanılan hazır veri setleri oluşturulmuştur.

STS'ler için veri setleri içerisinde en sık kullanılanlarından biri KDD Cup 99'dur. Aslında bu veri seti DARPA 1998 ve DARPA 1999 veri setlerinin başka bir versiyonudur. DARPA 1998 MIT Lincoln Laboratuvarlarında hazırlanmış bir veri setidir. Bu veri setleri Amerikan Hava Kuvvetlerinin ağ yapısına benzetilmeye çalışılarak tasarlanmıştır.

KDD Cup 99 veri seti 22 farklı saldırı türünde 41 özelliğe sahip yaklaşık 4 milyon data içermektedir. Veri setindeki kayıtlar 3 farklı protokole ait olabilmektedir. Bunlar TCP, UDP ve ICMP protokolleridir. Datalar saldırı olup olmadıklarına göre etiketlenmişlerdir. Bu saldırı türleri aşağıdaki tabloda yer almaktadır.

Tablo 3.1: Saldırı türleri.

back (dos)	ipsweep (probe)
buffer_overflow (u2r)	imap (r2l)
ftp_write (r2l)	loadmodule (u2r)
guess_passwd (r2l)	nmap (probe)
neptune (dos)	land (dos)
phf (r2l)	perl (u2r)
pod (dos)	warezmaster (r2l)
warezclient (r2l)	teardrop (dos)
spy (r2l)	multihop (r2l)
satan (probe)	smurf (dos)
portsweep (probe)	rootkit (u2r)

Daha önce belirtildiği gibi KDD Cup 99 veri setindeki her bir kayıt 41 özellikten oluşmaktadır ve bu kayıtların saldırı olup olmadıkları en sonda yer alan Class kolonunda belirtilmiştir. KDD Cup 99 veri setindeki özellikler sırasıyla aşağıdaki tabloda belirtilmiştir.

Tablo 3.2: Veri özellikleri.

1	Duration	22	is_guest_login
2	protocol_type	23	Count
3	Service	24	srv_count
4	Flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	Land	28	srv_error_rate
8	wrong_fragment	29	same_srv_rate
9	Urgent	30	diff_srv_rate
10	Hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login	42	Class

Bağlantı; iki IP arasında TCP paketlerinin belli bir zamanda ve tanımlanmış belli protokoller çerçevesinde taşınmasıdır. KDD içerisindeki özellikler 4 gruba ayrılmaktadır. Bunlar;

- **Temel özellikler:** Paket başlığında yer alan özelliklerdir.
- **İçerik özellikleri:** Paket parçalarının değerlendirilmesi için domain bilgisiyle ilgili olan alanlardır.
- **Zaman tabanlı trafik özellikleri:** Son 2 saniye içerisindeki bağlantıları inceler. Host ve servis olarak ikiye ayrılmaktadır. Host; 2 saniye içerisinde aynı hosta yapılan bağlantıları, servis; aynı servise ait bağlantıları ifade etmektedir.
- **Sunucu tabanlı trafik özellikleri:** Bazı probing saldırıları 2 saniyeden fazla süreyle portları ya da hostları tarayabilmektedir. Bu nedenle trafiği belli bir süre

için değil, aynı hosta yapılan belli bir bağlantı sayısı kadar incelemek gerekebilir. Bu özellikler aynı hosta yapılan 100 bağlantı üzerine kurgulanmıştır.

3.2. WEKA

WEKA, Waikato Üniversitesi tarafından makine öğrenmesi algoritmalarının gerçekleştirilmesi için oluşturulmuş bir yazılımdır. Java ile geliştirilmiş olduğu için Java entegrasyonu rahatça yapılabilmektedir. Veri madenciliği, makine öğrenmesi ve iş zekası gibi alanlarda en çok kullanılan yazılımlardandır. WEKA temel olarak gerçekleştirilebilecek 6 özellik mevcuttur. Bunlar;

- Sınıflandırma
- Kümeleme
- İlişkilendirme
- Veri Ön işleme
- Özellik seçimi
- Görselleştirme

Yazılım içerisinde çok sayıda hazır fonksiyon olduğu gibi Java dili aracılığıyla da kod yazılabilmektedir. WEKA kendine özel olan .arff uzantılı dosyalar ile çalışmaktadır.

3.2.1. Attribute-Relation File Format

Attribute-Relation File Format (ARFF) olarak bilinen dosya formatı WEKA için özel olarak geliştirilmiştir. Bu dosya içerisindeki data 3 farklı şekilde gruplanabilir. Bunlar;

- **İlişki satırı:** @relation ifadesi ile başlamaktadır. Bu ifadeden sonra veri kümesine verilmek istenen isim yazılmaktadır.
- **Özellik satırı:** @attribute ifadesi ile başlamaktadır. Bu ifadeden sonra veri kümesi içerisindeki özellik isimleri ve bunlara karşılık gelen veri tipleri yazılmaktadır.
- **Veri satırı:** @data ifadesi ile başlamaktadır. Daha sonraki satırlarda veri kümesi içerisindeki veriler yer alır.

.csv formatındaki dosyalar WEKA içerisinden Tools ->Arffviewer aracılığıyla .arff formatına dönüştürülebilmektedir.

Şekil 3.1: WEKA açılış ekranı.

3.3. YAZILIM ARAÇLARI

Verinin önişleme tabi tutulduğu kısımda Matlab kullanılarak veri sayısallaştırılmıştır. Versiyon olarak R2015b kullanılmıştır. Matlab'in tercih edilme sebebi performans olarak iyi çalışması ve hızlı sonuç üretmesinden kaynaklanmaktadır.

Sayısallaştırılan veri sonrasında WEKA aracılığıyla bir takım işlemlerden geçirilmiştir. YBS uygulamaları JAVA dili ile geliştirilmiştir. Versiyon olarak JAVA 7 kullanılmıştır. JAVA'nın tercih edilme sebebi WEKA ile entegrasyonun mevcut olması ve WEKA aracılığıyla üretilen veri kümesinin direkt kullanılabilecek olmasıdır. JAVA'nın kendi kütüphanelerine ek olarak WEKA için aşağıdaki kütüphaneler kullanılmıştır.

- weka.core.Instance
- weka.core.Instances
- weka.core.DenseInstance

4. BULGULAR

4.1. VERİ KÜMESİNİN DÜZENLENMESİ

KDD Cup 99 veri seti içerisinde string değerlerden oluşan toplam 4 kolon mevcuttur. Bunlar ikinci kolonda yer alan protocol_type, üçüncü kolonda yer alan service, dördüncü kolonda yer alan flag ve son kolonda saldırı türlerinin yer aldığı Class alanıdır. Veri kümesi üzerinde WEKA aracılığıyla özellik seçimi ve boyut indirgeme işlemlerinin yapılabilmesi için veriler sayısallaştırılmıştır. Son kolon dışında diğer string kolonların aldıkları değerler aşağıdaki tabloda yer alan nümerik değerler ile değiştirilerek tamamen sayısal değerlerden oluşan bir veri kümesi elde edilmiştir.

Tablo 4.1: Üçüncü kolonun alabileceği değerler.

1	http	17	Mtp	33	http_443	49	sunrpc	65	tim_i
2	smtp	18	Link	34	exec	50	uucp_path	66	red_i
3	domain_u	19	remote_job	35	printer	51	netbios_ns		
4	finger	20	gopher	36	efs	52	netbios_ssn		
5	auth	21	Ssh	37	courier	53	netbios_dgm		
6	telnet	22	name	38	uucp	54	sql_net		
7	ntp_u	23	whois	39	klogin	55	vmnet		
8	ftp	24	domain	40	kshell	56	bgp		
9	ecr_i	25	login	41	echo	57	Z39_50		
10	eco_i	26	imap4	42	discard	58	ldap		
11	other	27	daytime	43	systat	59	netstat		
12	private	28	Ctf	44	supdup	60	urh_i		
13	pop_3	29	nntp	45	iso_tsap	61	X11		
14	ftp_data	30	Shell	46	hostnames	62	urp_i		
15	rje	31	IRC	47	csnet_ns	63	pm_dump		
16	time	32	Nnsp	48	pop_2	64	tftp_u		

Tablo 4.2: İkinci kolonun alabileceği değerler.

1	tcp
2	udp
3	icmp

Tablo 4.3: Dördüncü kolonun alabileceği değerler.

1	SF
2	S0
3	S1
4	REJ
5	S2
6	S3
7	RSTO
8	RSTR
9	RSTOS0
10	OTH
11	SH

Sayısallaştırma işlemi sonrasında dataset içerisindeki gürültü oranını azaltmak için tekrarlayan veriler silinmiştir.

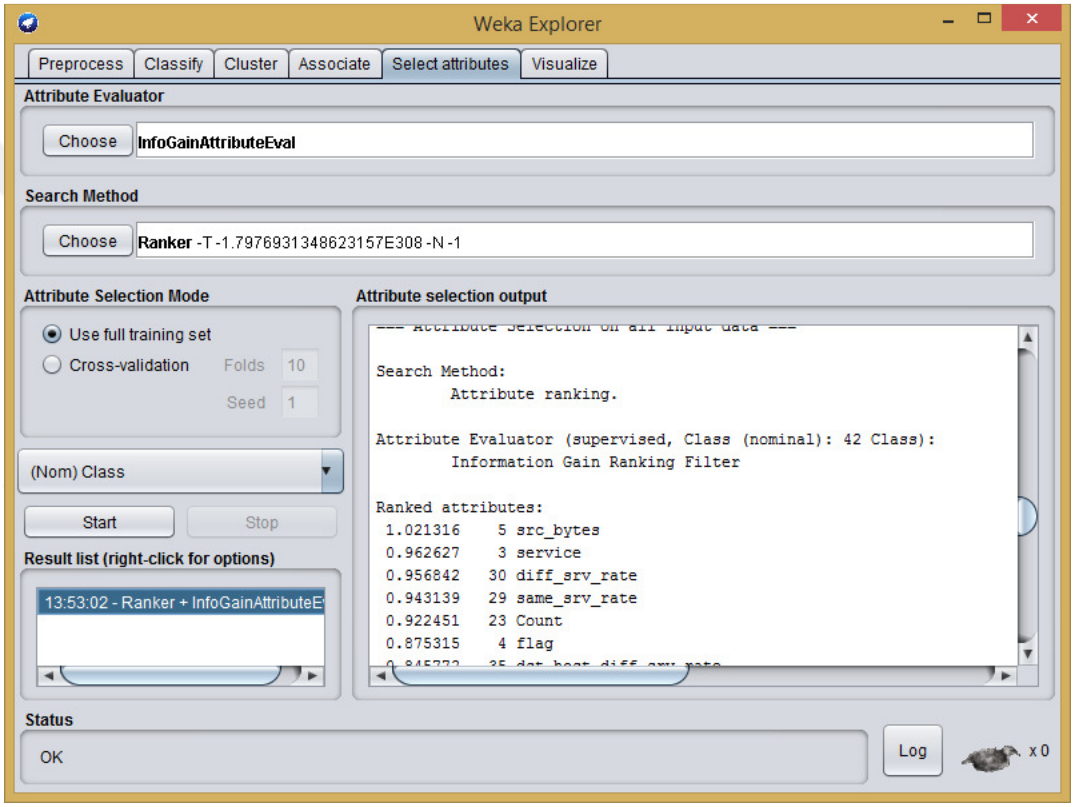
Özellik seçimi ve boyut indirgeme işlemi uygulanmadan gerçeklenen STS uygulaması için veriler üzerinde normalizasyon işlemi gerçekleştirilmiştir. Özelliklerin alabildikleri değerlerin dağılımları incelendiği zaman çok farklılık olduğu gözlemlenmiştir. Normalizasyon işlemi YBS içerisinde kullanılacak eşik değerinin belirlenmesi ve eğitim süresinin kısa olabilmesi için tercih edilmiştir.

Özellik seçimi ve boyut indirgeme işlemi uygulanacak STS'ler için veri kümesi normalizasyon işlemine tabi tutulmadan WEKA aracılığıyla .arff uzantılı yeni bir dosya oluşturulmuştur.

4.2. WEKA ARACILIĞIYLA ÖZELLİK SEÇİMİ VE TEMEL BİLEŞENLER ANALİZİ

Sayısallaştırılan veri kümesi üzerinde sistemin doğruluk değerinin arttırılması hedeflenerek özellik seçimi ve TBA uygulanmıştır. Özellik seçimi WEKA aracılığıyla 'InfoGainAttributeEval' tekniği kullanılarak yapılmıştır.

InfoGainAttributeEval yöntemi aracılığıyla her özelliğin bilgi kazancı değeri hesaplanmaktadır. Bilgi kazancı değerleri büyükten küçüğe olacak şekilde WEKA içerisinde sıralanmaktadır. Hesaplanan değerler içerisinde 9, 15, 20 ve 21. özelliklerin değerlerinin 0 çıktığı gözlemlenmiştir. Bir sonraki aşamaya geçmeden önce bu özellikler veri kümesinden çıkarılmıştır. <<InfoGainAttributeEval>> yöntemi sonrası oluşan ekran görüntüsü aşağıdaki şekildedir.



Şekil 4.1: WEKA özellik seçimi arayüzü.

Özellik seçimi işlemi sonrasında veri kümesi üzerinde WEKA aracılığıyla TBA uygulanmıştır. TBA uygulamaktaki amaç, doğruluk oranı ve karar verme hızı gibi faktörleri iyileştirmek iken aynı zamanda boyut indirgeme işlemi sonrasında azalacak özellik sayısının sistemin eğitilmesi sırasında kolaylık sağlamasıdır. TBA işlemi sonrası belirlenen öz değer ve öz vektörlere göre 21 TB üretilmiştir. Oluşturulan TB'lerin görüntüsü aşağıdaki şekildedir.

```
@attribute 0.328same_srv_rate+0.311dst_host_same_srv_rate-0.298dst_host_srv_serror_rate-0.298srv_serror_rate-0.298serror_rate... numeric
@attribute -0.427rerror_rate-0.426srv_error_rate-0.424dst_host_srv_rerror_rate-0.423dst_host_rerror_rate-0.331flag... numeric
@attribute 0.471protocol_type+0.466dst_host_same_src_port_rate+0.435dst_host_diff_srv_rate+0.347duration+0.181diff_srv_rate... numeric
@attribute 0.604num_compromised+0.604num_root+0.397num_access_files+0.277root_shell-0.102protocol_type... numeric
@attribute -0.687is_guest_login-0.685hot+0.096srv_count+0.086dst_host_srv_diff_host_rate+0.074protocol_type... numeric
@attribute 0.549srv_count-0.419dst_host_srv_diff_host_rate+0.369dst_host_count+0.258wrong_fragment-0.244srv_diff_host_rate... numeric
@attribute 0.471dst_host_srv_diff_host_rate+0.358srv_count-0.32dst_host_diff_srv_rate-0.306diff_srv_rate-0.283duration... numeric
@attribute -0.655num_shells-0.485num_file_creations-0.479root_shell+0.158num_compromised+0.158num_root... numeric
@attribute 0.681dst_bytes+0.669num_failed_logins-0.168num_shells-0.142num_file_creations-0.135srv_diff_host_rate... numeric
@attribute 0.998src_bytes-0.041wrong_fragment+0.035srv_count-0.024srv_diff_host_rate+0.016duration... numeric
@attribute 0.509num_file_creations-0.422wrong_fragment+0.393duration-0.333num_shells-0.323diff_srv_rate... numeric
@attribute -0.914land-0.231diff_srv_rate-0.127dst_host_count-0.126srv_count+0.118wrong_fragment... numeric
@attribute -0.682wrong_fragment-0.469num_file_creations+0.264num_shells+0.255srv_count-0.21and... numeric
@attribute 0.704num_failed_logins-0.689dst_bytes-0.105wrong_fragment+0.073diff_srv_rate-0.049num_file_creations... numeric
@attribute 0.503diff_srv_rate+0.479srv_diff_host_rate+0.393num_file_creations-0.338duration-0.257wrong_fragment... numeric
@attribute -0.709srv_diff_host_rate+0.378diff_srv_rate+0.297dst_host_srv_diff_host_rate-0.277duration-0.247dst_host_count... numeric
@attribute -0.732root_shell+0.501num_shells+0.412num_access_files-0.112srv_diff_host_rate+0.097num_failed_logins... numeric
@attribute -0.757num_access_files-0.311root_shell+0.298num_compromised+0.297num_root+0.219num_file_creations... numeric
@attribute -0.499service+0.358srv_count-0.323protocol_type+0.299duration+0.295wrong_fragment... numeric
@attribute 0.456dst_host_srv_diff_host_rate+0.432dst_host_count-0.388dst_host_same_src_port_rate+0.299duration+0.269protocol_type... numeric
@attribute -0.636service-0.455duration+0.305dst_host_diff_srv_rate+0.253dst_host_count-0.241diff_srv_rate... numeric
```

Şekil 4.2: KDD CUP'99 veri kümesi temel bileşenleri.

4.3. EĞİTİM VE TEST VERİ KÜMELERİ

Literatürdeki çalışmalar incelendiğinde veri kümelerinin farklı örnek miktarlarında ve farklı oranlarda oluşturulduğu gözlemlenmiştir. Bazı çalışmalarda eğitim veri kümesindeki datalar az tutulurken bazılarında ise test veri kümelerindeki örnekler daha az tutulmuştur.

Bu çalışma kapsamında ön işlemler tamamlandıktan sonra veri kümesi eğitim ve test aşamalarında kullanılmak üzere iki farklı kümeye bölünmüştür. Bu kapsamda eğitim ve test veri kümeleri içerisindeki normal ve saldırı kayıtlarının sayısı aşağıdaki tabloda paylaşılmıştır.

Tablo 4.4: Veri kümelerindeki örnek sayısı dağılımı.

Veri kümesi	Normal örnek sayısı	Saldırı örnek sayısı
Eğitim veri kümesi	57975	38111
Test veri kümesi	29857	19643

4.4. TESTLER SIRASINDA HESAPLANACAK PARAMETRELER

Makine öğrenmesi sonucunda oluşturulan modellerin değerlendirilmesi için farklı yöntemler mevcuttur. Model üzerinde bir değerlendirme yapılmadan gerçek hayatta kullanılması mümkün değildir. Tez kapsamında geliştirilen YBS algoritmaları ve

WEKA aracılığıyla kullanılan makine öğrenmesi algoritmaları için aşağıdaki değerler hesaplanacaktır.

- **Doğruluk:** Bir testin doğruluğu pozitif ve negatif değerlerin doğru bir şekilde ayırt edilmesi ile ölçülmektedir. Doğruluk oranı hesaplanırken doğru karar verilen normal kayıtların sayısı ve doğru karar verilen saldırı kayıtlarının sayısı toplanarak toplam örnek sayısına bölünür.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{\text{DP} + \text{YP} + \text{DN} + \text{YN}} \tag{4.1}$$

- **Duyarlılık:** Veri kümesi içerisinde normal olan kayıtlar için sistemin normal olarak karar verdiği örnek sayısının toplam normal sayısına bölünmesi ile edilir.

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP} + \text{DN}} \tag{4.2}$$

- **Seçicilik:** Veri kümesi içerisinde saldırı olan ve sistemin saldırı olarak karar verdiği örnek sayısının toplam saldırı sayısına bölünmesi ile elde edilir.

$$\text{Seçicilik} = \frac{\text{DN}}{\text{DN}+\text{YP}} \tag{4.3}$$

  Bir testin doğruluğundan bahsedebilmek için hem duyarlılık hem de seçicilik değerlerinin yüksek olması gerekmektedir [54].

- **Kesinlik:** Veri kümesi içerisinde normal olup normal olarak sınıflandırılan örnek sayısının, normal olarak sınıflandırılan tüm örnek sayısına bölünmesi ile elde edilir.

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP}+\text{YP}} \tag{4.4}$$

- **F-ölçütü:** Kesinlik ve duyarlılık parametrelerinin tek başına sınıflandırma için ölçüt olarak gösterilmesi yeterli olmamaktadır. F-ölçütü bu iki parametrenin birlikte kullanılması ile hesaplanan ve sınıflandırma için daha anlamlı çıkarım

yapılmasına yarayan bir parametredir. F-ölçütü kesinlik ve duyarlılığın harmonik ortalaması olarak ifade edilmektedir.

$$\text{F-ölçütü} = \frac{2 * \text{Duyarlılık} * \text{Kesinlik}}{\text{Duyarlılık+Kesinlik}} \quad (4.5)$$

Bu değerlerin hesaplanması için kullanılacak parametreler karışıklık matrisi denilen, sınıflandırma sistemi içerisindeki gerçek ve tahmin edilen sonuçları gösteren bir matriste yer almaktadır. İki sınıflı bir örnek için karışıklık matrisi aşağıdaki gibidir.

Tablo 4.5: Karışıklık Matrisi.

		TAHMİN EDİLEN		TOPLAM
		Doğru	Yanlış	
GERÇEK	Doğru	DP Doğru Pozitif	YN Yanlış Negatif	Gerçek Pozitif Sayısı
	Yanlış	YP Yanlış Pozitif	DN Doğru Negatif	Gerçek Negatif Sayısı
TOPLAM		Tahmin Pozitif Sayısı	Tahmini Negatif Sayısı	Toplam Örnek Sayısı

4.5. YAPAY BAĞIŞIKLIK SİSTEMİ ALGORİTMASININ GERÇEKLENMESİ

Bu çalışma kapsamında YBS algoritmalarından çıkış amacı ağ güvenliği olan negatif seçim algoritması kullanılmıştır. Makine öğrenmesi algoritmalarının gerçeklenmesi 3 aşamadan oluşmaktadır. Bunlar aşağıdaki gibidir.

- Verinin düzenlenmesi
- Eğitim Aşaması
- Test Aşaması

Kullanılacak verinin düzenlenmesi ile ilgili kısım daha önceki bölümlerde anlatılmıştır. Yapay bağışıklık algoritmaları kapsamında eğitim aşaması detektör üretimine denk gelmektedir. Test aşamasında üretilen detektörler ile test veri kümesi karşılaştırılarak eşik değerleri kapsamında vektörlerin saldırı olup olmadıklarına karar verilmektedir.

Çalışma kapsamında 3 farklı STS uygulaması geliştirilmiştir. Bunlar aşağıdaki gibidir.

- Özellik seçimi ve boyut indirgeme uygulanmadan gerçekleştirilen STS (Rastgele detektör üretimi yöntemi kullanılmıştır.) – YBS-1
- Özellik seçimi ve boyut indirgeme uygulanarak gerçekleştirilen STS (Rastgele detektör üretimi yöntemi kullanılmıştır.) – YBS-2
- Özellik seçimi ve boyut indirgeme uygulanarak gerçekleştirilen STS (GA ile detektör üretimi yöntemi kullanılmıştır.) – YBS-3

Tüm yöntemler için eğitim ve test aşamalarında aynı örnekler kullanılmıştır.

4.5.1. Detektör Üretimi

Bu çalışma kapsamında detektör üretimi için iki farklı yöntem kullanılmıştır. Bunlardan ilki negatif seçim algoritması için geleneksel olan rastgele detektör üretimi iken diğer yöntem GA aracılığıyla detektör üretimidir. GA kullanılmasının sebebi sistemi rastgelelikten kurtararak en iyi çözüme belli prosedürler çerçevesinde ulaşmak ve bunun sistemin karar vermesine etkisini gözlemlemektir.

Her iki yöntem için de benzerlik hesaplaması yapılırken Öklid mesafesi kullanılmıştır. Rastgele detektör üretimi için her kolonun alabildiği minimum ve maksimum değerler hesaplanmış, bu değerler arasında olacak şekilde yeni kolon değerleri üretilmiştir. Her kolon için bu işlem tekrarlanarak bir vektör elde edilmiş ve bu vektör eğitim veri kümesi içerisinde yer alan normal örnekler ile karşılaştırılmıştır. Hesaplanan Öklid değeri sonrasında yapılan karşılaştırmaya göre üretilen vektörün detektör olup olmadığına karar verilmiştir.

GA ile detektör üretimi aşamasında hem normal hem de saldırı örnekleri kullanılmıştır. Detektör üretimi için sırasıyla aşağıdaki adımlar gerçeklenmiştir.

- Normal ve saldırı örneklerinden oluşacak şekilde rastgele popülasyon kümesi üretilmiş ve bireyler için uygunluk değeri hesaplanmıştır.
- Uygunluk değerinin hesaplanması için Öklid mesafesi formülü kullanılmıştır.
- Ebevyn olarak seçilecek bireylerden biri normal, biri saldırı örneklerinden olacak şekilde rulet çarkı yöntemi kullanılarak seçilmiştir.
- Elde edilen iki ebevyn üzerinde çaprazlama işlemi uygulanmıştır. Çaprazlama yöntemi olarak tek noktalı çaprazlama tekniği kullanılmıştır.

- Oluşturulan yeni bireyler üzerinde mutasyon işlemi yapılmıştır. Kullanılan değerler gerçek sayılardan oluştuğu için 0 ve 1 aralığında rastgele bir sayı üretilerek bu değer yine rastgele belirlenen bir gene eklenmiştir.
- Mutasyon işlemi sonrası oluşturulan yeni bireyler için tekrar uygunluk değeri hesaplanarak daha önce üretilen detektörler ile karşılaştırılmıştır. Eğer kendilerinden daha büyük uygunluğa (Öklid mesafesi) sahip bireyler bulunursa bunlarla yer değiştirilerek detektör kümesine yeni bireyler dahil edilmiştir.

4.6. TEST SONUÇLARI VE YÖNTEMLERİN KARŞILAŞTIRILMASI

4.6.1. Senaryo 1 : YBS-1 için Test Sonuçları

YBS-1 algoritması geliştirilen hibrid yöntemlerdeki sonuçların değerlendirilmesi için geliştirilmiş bir uygulamadır. Bu yöntem kapsamında veri kümesi içerisindeki 41 özellik de kullanılmıştır. YBS-1 algoritmasının detektör sayısı bazındaki test sonuçları aşağıdaki gibidir.

Tablo 4.6: YBS-1 algoritmasının detektör sayısı bazında test sonuçları.

Detektör Sayısı	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
50	92.75	93.72	91.29	94.24	93.98
100	96.2	99.69	90.9	94.33	96.94
150	93.28	94.36	91.64	94.49	94.42
200	90.13	87.39	94.3	95.89	91.44
250	90.11	92.04	87.19	91.61	91.82

4.6.2. Senaryo 2 : Temel Bileşen Sayısı Bazında Test Sonuçları

Uygulamanın test aşamasında normal ve saldırı örneklerinden oluşan test veri kümesi kullanılmıştır. YBS-2 ve YBS-3 algoritmaları için öncelikle TB bazında test gerçekleştirilerek kullanılacak TB sayısına karar verilmiştir. Her iki algoritma için ayrı ayrı 21 (toplam TB sayısı) kez test gerçekleştirilmiştir. Her TB için daha önce bahsedilen performans parametreleri hesaplanmıştır. TB'ler bazında yapılan testler için detektör sayısı ve popülasyon sayısı 100 olarak kabul edilmiştir.

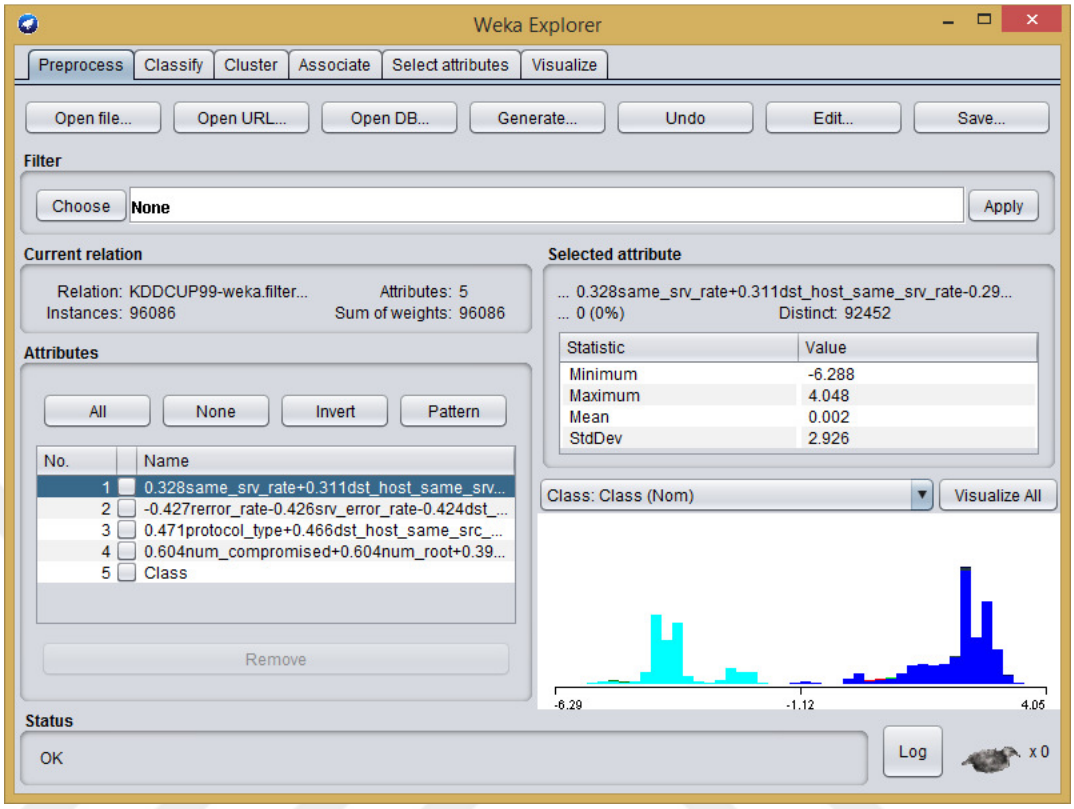
YBS-2 algoritması için hesaplanan performans parametreleri TB bazında Tablo 4.7'de detaylı bir şekilde paylaşılmıştır.

Tablo 4.7: Rastgele detektör üretimi için temel bileşen bazında test sonuçları.

Temel Bileşen Sayısı	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
1	92.53	90.84	95.11	96.58	93.62
2	95.17	96.76	92.75	95.3	96.03
3	94.95	96.1	93.19	95.55	95.82
4	98.5	99.37	97.18	98.17	98.76
5	96.05	98.05	93.02	95.52	96.77
6	88.13	98.52	72.34	84.41	90.92
7	90.61	99.74	76.73	86.69	92.76
8	91.18	93.33	87.89	92.14	92.73
9	86.93	94.24	75.81	85.55	89.69
10	92.74	94.66	89.81	93.39	94.02
11	88.86	94.1	80.9	88.22	91.07
12	90.22	92.1	87.36	91.72	91.91
13	88.54	96.53	76.4	86.14	91.04
14	86.09	90.62	79.21	86.88	88.71
15	90.68	94.35	85.1	90.59	92.43
16	89.68	91.27	87.27	91.59	91.43
17	89.16	88.45	90.23	93.22	90.77
18	88.44	86.78	90.96	93.59	90.05
19	90.83	91.73	89.47	92.98	92.35
20	81.84	77.15	88.96	91.4	83.67
21	88.21	88.22	88.2	91.91	90.03

Yukarıdaki tablodan görülebileceği gibi YBS-2 için en iyi sonuçlar TB sayısının 4 olduğu durumda elde edilmiştir. Sonrasında doğruluk değerleri giderek azalmıştır.

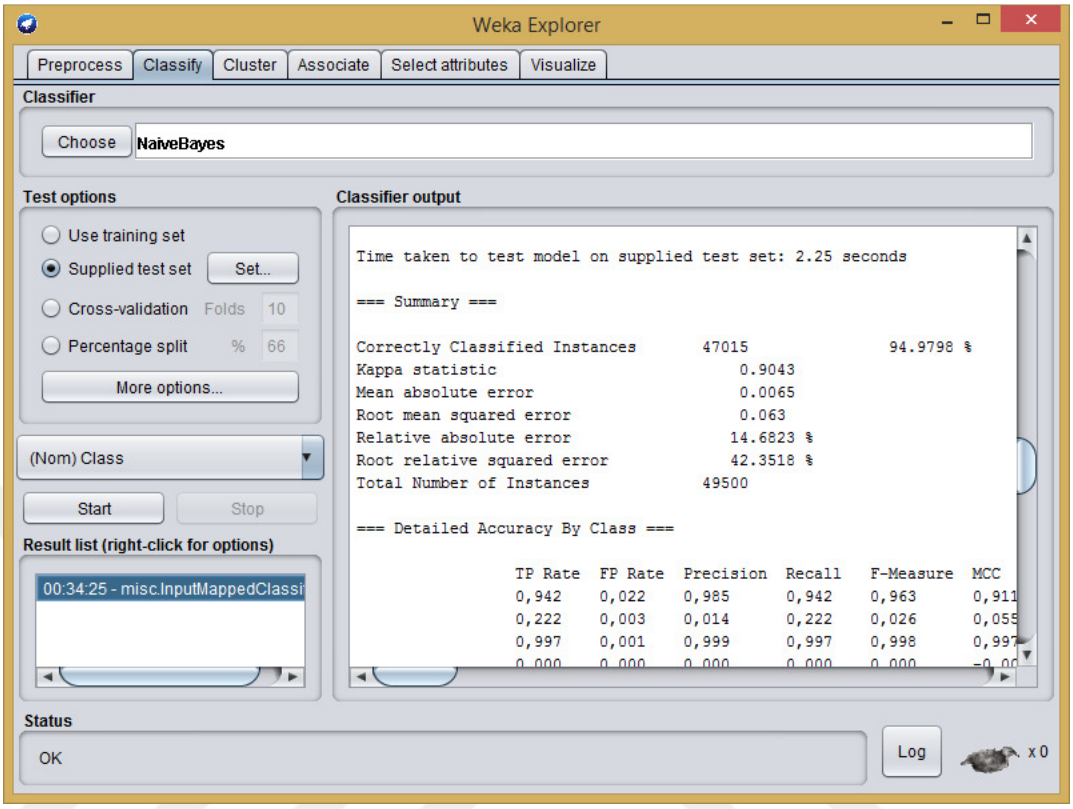
YBS-2 için TB sayısı 4 seçildikten sonra elde edilen sonuçlar WEKA içerisinden seçilen bazı makine öğrenmesi algoritmaları ile karşılaştırılmıştır. WEKA içerisine eğitim veri kümesi yüklendikten sonra ilk dört özellik dışındaki özellikler silinmiştir. Silme işlemi WEKA'nın Preprocess sekmesinde sol attaki Remove butonu kullanılarak gerçekleştirilmiştir.



Şekil 4.3: Özellik silme işlemi sonrasında veri gösterimi.

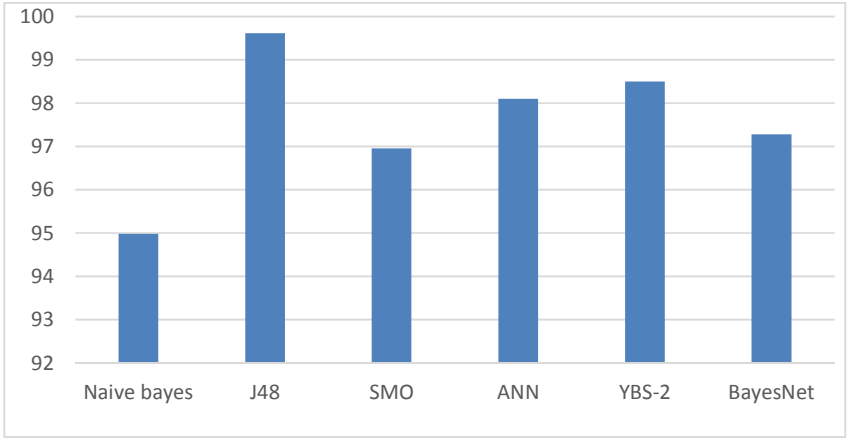
Özellik silme işlemi sonrasında Classify sekmesine geçilerek kullanılacak sınıflandırma algoritmaları seçilmiştir. WEKA içerisinde algoritma seçildikten sonra test için kullanılacak veri kümesi veya yöntemler ile ilgili seçim yapılabilmektedir. Supplied test set seçeneği kullanılarak test aşamasında kullanılacak yeni bir veri kümesi seçilerek belirlenen algoritmalar çalıştırılmıştır. WEKA içerisinden kullanılan sınıflandırma algoritmaları aşağıdaki gibidir.

- Naive Bayes
- J48
- SMO
- ANN
- BayesNet



Şekil 4.4: Naive Bayes algoritmasının sonuçları.

Tüm algoritmaların gerçeklenmesinden sonra doğruluk oranları aşağıda yer alan grafiğe aktarılmıştır. YBS-2'nin doğruluk oranı diğer algoritmalara göre oldukça iyi durumdadır.



Şekil 4.5: YBS-2 ve WEKA algoritmaları karşılaştırması.

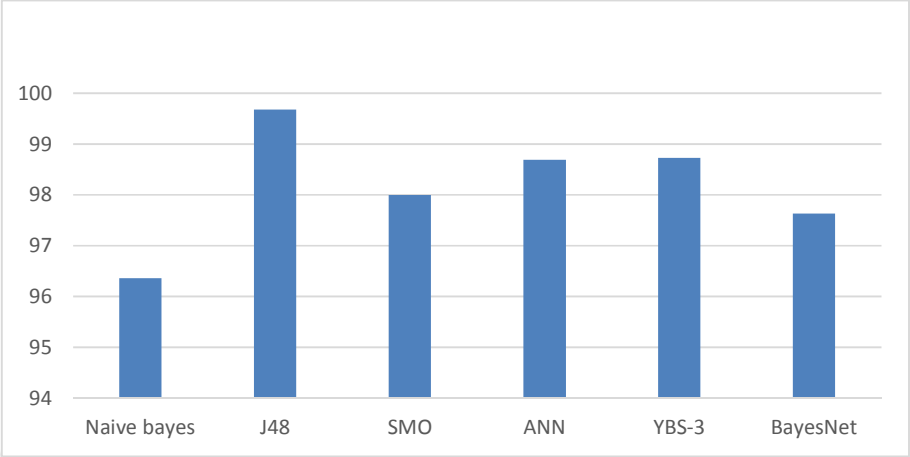
YBS-3 algoritması için hesaplanan performans parametreleri TB bazında Tablo 4.8'de detaylı bir şekilde paylaşılmıştır. YBS-2 ve YBS-3 karşılaştırıldığında YBS-3'ün hedeflendiği gibi daha istikrarlı sonuçlar ürettiği gözlemlenmiştir.

Tablo 4.8: Genetik algoritma ile detektör üretimi için temel bileşen bazında test sonuçları.

Temel Bileşen Sayısı	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
1	91.21	88.28	95.66	96.87	92.37
2	96.01	99.71	90.38	94.03	96.79
3	96.09	97.55	93.87	96.03	96.78
4	96.41	98.5	93.23	95.68	97.07
5	96.08	97.12	94.5	96.41	96.76
6	96.59	96.93	96.09	97.41	97.17
7	97.24	98.12	95.89	97.32	97.72
8	96.53	96.96	95.88	97.28	97.12
9	96.74	97.05	96.26	97.53	97.29
10	96.71	96.97	96.3	97.55	97.26
11	97.79	98.37	96.91	97.98	98.17
12	98.73	99.2	98.01	98.69	98.95
13	97.2	97.88	96.18	97.49	97.69
14	97.31	98.12	96.06	97.43	97.78
15	96.84	97.33	96.09	97.43	97.38
16	97.08	97.52	96.41	97.64	97.58
17	97.12	97.79	96.09	97.44	97.61
18	96.61	96.66	96.53	97.69	97.17
19	97.53	98.6	95.92	97.35	97.97
20	97.73	99.03	95.75	97.26	98.14
21	95.87	96.26	95.29	96.88	96.57

Yukarıdaki tablodan görülebileceği gibi YBS-2 için en iyi sonuçların alındığı TB sayısı 4 çıkarken YBS-3 için 12 çıkmıştır. YBS-2 deki azalma oranından daha iyi olmasına karşın YBS-3 için de 12'den sonra doğruluk değerleri giderek azalmıştır.

YBS-3 için TB sayısı 12 seçildikten sonra elde edilen sonuçlar tekrar WEKA içerisinden seçilen bazı makine öğrenmesi algoritmaları ile karşılaştırılmıştır. WEKA içerisinden kullanılan sınıflandırma algoritmaları YBS-2 için kullanılanların aynısıdır. Tüm algoritmaların tekrar gerçeklenmesinden sonra doğruluk oranları aşağıda yer alan grafiğe aktarılmıştır. YBS-3'ün doğruluk oranı YBS-2'de olduğu gibi diğer algoritmalara göre oldukça iyi durumdadır.



Şekil 4.6: YBS-3 ve WEKA algoritmaları karşılaştırması.

4.6.3. Senaryo 3 : Detektör Sayısı Bazında Test Sonuçları

Senaryo 2 kapsamında yapılan testler sonrasında en iyi sonuçların elde edildiği TB sayısı bulunmuştur. Senaryo 3'de ise bu TB sayılarına göre detektör sayıları değiştirilerek bunların sistemin karar vermesine olan etkisi gözlemlenmiştir. Modelin oluşturulması ve test edilmesi aşamasında kullanılan veri kümeleri Senaryo 2'deki ile aynıdır.

YBS-2 için TB sayısı 4 seçilerek yapılan testlerin sonucu Tablo 4.9'da paylaşılmıştır. Sonuçlara göre oranlar birbirine oldukça yakındır. En yüksek değerlerin elde edildiği aşama detektör sayısının 100 olduğu kısımdır.

Tablo 4.9: YBS-2 algoritmasının detektör sayısı bazında test sonuçları.

Detektör Sayısı	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
50	94.08	96.13	90.96	94.17	95.14
100	98.5	99.37	97.18	98.17	98.76
150	96.28	99.33	91.64	94.75	96.99
200	96.02	99.37	90.93	94.34	96.79
250	96.12	99.87	90.41	94.06	96.88

YBS-2'de olduğu gibi YBS-3 için de TB sayısı sabit tutularak detektör sayısına göre testler yapılmış ve sonuçları Tablo 4.10'da paylaşılmıştır. YBS-3 için detektör sayısının değişimine ek olarak popülasyon sayısı da detektör sayısı kadar arttırılmıştır. En yüksek değerlerin elde edildiği aşama detektör sayısının 100 olduğu kısımdır.

Tablo 4.10: YBS-3 algoritmasının detektör sayısı bazında test sonuçları.

Detektör Sayısı	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
50	97.1	97.87	95.92	97.33	97.6
100	98.73	99.2	98.01	98.69	98.95
150	96.4	96.47	96.29	97.53	97
200	97	98.06	95.39	97	97.53
250	97.52	98.43	96.12	97.47	97.95

5. TARTIŞMA VE SONUÇ

Çalışmanın ilk kısmında sistemin karar verme yeteneğini ve performansını güçlendirecek metotlar araştırılmıştır. Araştırmalar sonucunda özellik seçimi ve boyut indirgeme ile ilgili yöntemlerin kullanılmasına karar verilmiş ve veri sayısallaştırılarak bu işlemler için hazır hale getirilmiştir.

Özellik seçimi ve boyut indirgeme işlemleri için makine öğrenmesi uygulamalarında sıklıkla kullanılan WEKA yazılımı kullanılmış ve bu yazılımın çalışma şekli incelenmiştir. Bu işlemler sonrasında üretilen .arff uzantılı yeni bir dosya ve tamamen reel değerlerden oluşan bir veri kümesi elde edilmiştir.

Çalışmanın ikinci kısmında yapay bağışıklık algoritmaları JAVA diliyle gerçeklenmiştir. Yine bu kısımda üretilecek olan sonuçların rastgelelikten kurtarılması için hibrid bir sistem uygulanmıştır. Negatif seçim algoritmasının eğitim aşaması olan detektör üretimi, hem rastgele hem de GA'lar aracılığıyla geliştirilerek elde edilen sonuçlara etkileri gözlemlenmiştir. GA'nın sağladığı en iyi çözümü üretme ilkesinden dolayı rastgele detektör üretimine göre daha sağlıklı sonuçlar elde edilmiştir.

Çalışmanın son kısmında geliştirilen yöntemlerin başka makine öğrenmesi algoritmaları ile karşılaştırılmaları hedeflenmiş ve bu kapsamda WEKA içerisindeki bir takım algoritmalar kullanılmıştır. Hem YBS-2 hem de YBS-3 algoritmaları diğer algoritmalar karşısında oldukça tatmin edici sonuçlar elde etmişlerdir.

YBS-1 ve diğer algoritmalar karşılaştırıldığı zaman çalışmanın başında hedeflenen doğruluk oranlarının arttırılması kriterinin gerçeklenmiş olduğu gözlemlenmiştir. Özellik seçimi ve boyut indirgeme işlemleri uygulanmadan gerçekleştirilen YBS-1 algoritmasının doğruluk oranı %96.2 iken bu işlemler uygulandıktan sonra gerçekleştirilen YBS-2 algoritması %98.5 doğruluk oranına ulaşmıştır. Son olarak sistemi rastgelelikten kurtarmak için gerçekleştirilen YBS-3 algoritmasının doğruluk oranı %98.73'e ulaşmıştır. Aynı zamanda YBS-3'ün sonuçları incelendiği zaman daha istikrarlı sonuçlar ürettiği de gözlemlenmiştir.

İlerideki çalışmalarda algoritmalar üzerinde iyileştirme yapılması ve benzerlik ölçümü, özellik seçimi gibi yöntemler için farklı metotların denenmesi hedeflenmektedir. Ayrıca

saldırının tespitinin dışında veri ve özelliklerin kullanılması sonucu saldırı türünün ne olduğuyla ilgili tahminlerin yapılması da ayrı bir çalışma konusu olarak düşünülmektedir.

KAYNAKLAR

[1]. Bharti, K. and Jain, S., Shukla, S., 2010, Fuzzy K-mean Clustering Via J48 For Intrusion Detection System, *International Journal of Computer Science and Information Technologies*, 1 (4) , 315-318.

[2]. Jha, J. and Ragha, L., 2013, Intrusion Detection System using Support Vector Machine, *International Journal of Applied Information Systems*, ISSN : 2249-0868.

[3]. Mukkamala, S., Sung, A. and Abraham, A., 2004, Intrusion Detection using an ensemble of intelligent paradigms, *Journal of Network and Computer Applications*, 28 (2005), 167-182.

[4]. Venkatachalam, V. and Selvan, S., 2007, Intrusion Detection using an Improved Competitive Learning Lamstar Neural Network, *IJCSNS International Journal of Computer Science and Network Security*, 7 (2), 255-263.

[5]. Aziz, A., Salama, M., Hassanien, A. and Hanafi, S., 2012, Detectors generation using genetic algorithm for a negative selection inspired anomaly network intrusion detection system, *Federated Conference on Computer Science and Information Systems*, 9-12 September 2012 Poland, IEEE, ISBN : 978-1-4673-0708-6, 597-602.

[6]. Dasgupta, D., 1998, *An Overview of Artificial Immune Systems and Their Applications*, Artificial Immune Systems and Their Applications, In: Dasgupta, D.(ed.), Chapter 1, Springer-Verlag Berlin Heidelberg, Berlin, Germany, 3-21.

[7]. Aydın, M.A., 2005, *Bilgisayar Ağlarında Saldırı Tespiti için İstatistiksel Yöntem Kullanılması*, Thesis (MSc), Institute of Science and Technology, Istanbul Technical University.

[8]. Pawar, M. and J., Anuradha, 2015, Network Security and Types of Attacks in Network, *International Conference on Intelligent Computing, Communication & Convergence* , 3-5 December 2015 Bhubaneswar, Odisha, India, 503-506.

[9]. Daya, B., 2013, Network Security: History, Importance, and Future, University of Florida Department of Electrical and Computer Engineering.

[10]. Erol, M., 2005, Saldırı Tespit Sistemlerinde İstatistiksel Anormallik Belirleme Kullanımı, Istanbul Technical University, Istanbul.

[11]. Paliwal, S. and Gupta, R., 2012, Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm, *International Journal of Computer Applications*, 60 (19), 57-62.

[12]. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A., 2009, A Detailed Analysis of the KDD CUP 99 Data Set, *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 8-10 July 2009 Canada, 1-6.

[13]. Bace, R. and Mell, P., 2001, *Intrusion Detection Systems*, Gaithersburg, USA, ISBN: 800-31.

[14]. Bunel, P., 2004, An Introduction to Intrusion Detection Systems, *SANS Conference*, June 2004 London, GIAC Security Essentials Certification.

[15]. Cracknell, M.J. and Reading, A.M., 2013, Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information, *Computers & Geosciences*, 63 (2014), 22-33.

[16]. Emel, G.G. and Taşkın, Ç., 2002, Genetik algoritmalar ve uygulama alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (1), 129-152.

[17]. Shukla, A., Pandey, H. and Mehrotra, D., 2015, Comparative Review of Selection Techniques in Genetic Algorithm, *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management*, 25-27 February India, 515-519.

[18]. Kumar, R., 2012, Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms, *International Journal of Machine Learning and Computing*, 2 (4), 365-370.

[19]. Umbarkar, A.J. and Sheth, P.D., 2015, Crossover Operations in Genetic Algorithms: A Review, *ICTACT Journal on Soft Computing*, 6 (1), 1083-1092.

[20]. Soni, N. and Kumar, T., 2014, Study of Various Mutation Operators in Genetic Algorithms, *International Journal of Computer Science and Information Technologies*, 5 (3), 4519-4521.

[21]. Bıyıklıoğlu, O., 2004, *Use of Artificial Immune Systems for Network Intrusion Detection*, Thesis (MSc), Institute of Science and Technology, İstanbul Technical University.

[22]. Castro, L.N. and Timmis, J., 2002, Artificial Immune Systems: A Novel Paradigm to Pattern Recognition, *In Artificial Neural Networks in Pattern Recognition,* 67-84.

[23]. Diao, Y. and Passino, K., 2002, Immunity-based hybrid learning methods for approximator structure and parameter adjustment, *Engineering Applications of Artificial Intelligence,* 15 (6), 587-600.

[24]. Castro, L.N., 2002, Immune, Swarm and Evolutionary Algorithms Part I: Basic Models, *International Conference on Neural Information Processing*, 18-22 November Singapura, 1464-1468.

[25]. Pamuk, Z., 2014, *Yapay Bağışıklık Sistemine Dayalı Yeni Bir Aritmi Sınıflama Tekniği*, Thesis (PhD), Fen Bilimleri Enstitüsü, Sakarya Üniversitesi.

[26]. Yurttakal, A.H., 2014, *İş Akışı Çizelgeleme Probleminin Yapay Bağışıklık Sistemi ile Optimizasyonu*, Thesis (MSc), Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

[27]. Bircan, Ç., 2014, *Yapay Bağışıklık Algoritmasına Dayalı Yeni Bir Bulanık Zaman Serisi Çözüm Yöntemi*, Thesis (MSc), Fen Bilimleri Enstitüsü, Ondokuz Mayıs Üniversitesi.

[28]. Bayğın, M. and Karaköse, M., 2011, Adaptif Yapay Bağışık Sistem Tabanlı Grup Asansör Kontrol Algoritması, *Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 5-7 Ekim 2011 Elazığ, 205-210.

[29]. Forrest, S., Perelson, A.S., Allen, L., and Cherukuri., R., 1994, Self-Nonself Discrimination in a Computer, *In Proceedings of IEEE Symposium on Research in Security and Privacy*, 16-18 May 1994 Oakland , 202- 212.

[30]. Dasgupta, D., 2012, Immunity-Based Intrusion Detection System: A General Framework, *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications,* 2012 India, Springer India, 417-428.

[31]. Düzdar, I. and Temür, G., 2017, Veri Madenciliği Yazılımlarının Kümeleme Yöntemlerinin Bir Veri Seti ile Gerçeklenmesi ve Sonuçlarının Karşılaştırılması, *Elektrik-Elektronik, Bilgisayar, Biyomedikal Mühendislikleri Bilimsel Toplantısı,* 20-21 April 2017 İstanbul.

[32]. Dasgupta, D. and Ji, Z., 2007, Revisiting Negative Selection Algorithm, *Evolutionary Computation,* 15 (2), 223-251.

[33]. Aydın, I., Karaköse, M. and Akın E., 2009, Genetik Algoritma Kullanan Yapay Bagısık Sistem Tabanlı Arıza Teshis Modeli, *Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi*, 11 (31), 57-72.

[34]. Burnet, F.M., 1959, The Clonal Selection Theory of Acquired Immunity, Cambridge University Press.

[35]. Castro, L.N., and Von Zuben, F.J., 2000, The Clonal Selection Algorithm with Engineering Applications, *Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, 36-37.

[36]. Polat, K., 2004, *Özellik seçme (FS) ile yapay bağışıklık tanıma sistemi (AIRS) kullanılarak medikal teşhise gidiş*, Thesis (MSc), Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

[37]. Engin, O. and Döyen, A., 2004, Artificial Immune Systems and Applications in Industrial Areas, *G.U. Journal of Science*, 17 (1), 71-84.

[38]. Yazıcı, B., Yaslı, F., Gürleyik, Y.H., Turgut, U., Aktas, M. and Kalıpsız, O., 2015, Veri Madenciliğinde Özellik Seçim Tekniklerinin Bankacılık Verisine Uygulanması Üzerine Araştırma ve Karşılaştırmalı Uygulama, *Ulusal Yazılım Mühendisliği Sempozyumu,* 9-11 September 2015 İzmir, 72-83.

[39]. Kaliyeva, S., 2013, *Bilimsel Makalelerin Metin İşleme Yöntemleri ile Sınıflandırılması*, Thesis (MSc), Fen Bilimleri Enstitüsü, Gazi Üniversitesi.

[40]. Guyon, I. and Elisseeff, A., 2003, An Introduction to Variable and Feature Selection, *Journal of Machine Learning Research*, 3 (2003), 1157-1182.

[41]. Yildirim, P., 2015, Filter Based Feature Selection Methods for Prediction of Risks in Hepatitis Disease, *International Journal of Machine Learning and Computing*, 5 (4), 258-263.

[42]. Chen, W., Er, M. and Wu, S., 2005, PCA and LDA in DCT domain, *Pattern Recognition Letters*, 26 (15), 2474–2482.

[43]. Abdi, H., Williams, L.J., 2010, Principal Component Analysis, *Wiley Interdisciplinary Reviews: Computational statistics*, 2 (4), 433-459.

[44]. Vasan, K.K. and Surendiran, B., 2016, Dimensionality reduction using Principal Component Analysis for network intrusion detection, *Perspective in Science*, 2016 (8), 510-512.

[45]. Çetiner, H., Kuşcu, Ö. and Tatlı, M., 2014, Boyutu Yüksek Görüntülerin Öznitelik Dönüşüm Yöntemiyle Analizi, *Akademik Bilişim Konferansı Bildirileri*, 5-7 February 2014 Mersin, 907-914.

[46]. Smith, L., 2002, *A tutorial on Principal Component Analysis*, Otago University, New Zealand, 1–26.

[47]. Alpaydın, E., 2004, *Introduction to Machine Learning*, The MIT Press, London, ISBN: 9780262325738.

[48]. Sridevi, R., Jagajothi, G. and Chattemvelli, R., 2012, A PCA-AIS Approach for Intrusion Detection, *International Journal of Computer Science and Telecommunications*, 3 (7), 104-108.

[49]. Nabilah, H.E., Wan, K., Talha, S.K., Ali, M.H., Shahriman, A., Razlan, M.Z. and Hazry, D., 2015, Recognition of Objects by Grasping Force Using Linear Discriminant Analysis (LDA), *International Conference on Control System, Computing and Engineering*, 27 - 29 November 2015 Malaysia, 254-259.

[50]. Martinez, M.A. and Kak, A.C., 2001, PCA versus LDA, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (2), 228-233.

[51]. Datti, R. and Verma, B., 2010, Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis, *International Journal on Computer Science and Engineering*, 2 (4), 1072-1078.

[52]. Balakrishnama S. and Ganapathiraju, A., Linear Discriminant Analysis - A Brief Tutorial, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University.

[53]. Shereena, V.B. and David, J.M., 2015, Significance of Dimensionality Reduction in Image Processing, Signal & Image Processing : An International Journal (SIPIJ), 6 (3), 27-42.

[54]. Stojanovic, M., Apostlovic, M., Stojanovic, D., Milosevic, Z., Toplaovic, A., Lakusic,V. and Golubovic, M., 2014, Understanding sensitivity, specificity and predictive values, *Vojnosanit Pregl*, 71 (11), 1062–1065.

EKLER

EK 1. Eğitim kümesi olarak kullanılan trainset.arff dosyasının ilk sayfası aşağıdaki gibidir.

@data

2.215798,0.472487,-0.269866,0.016854,-0.042286,-0.508187,-0.21719,-0.022472,0.158388,-0.002277,-0.020375,0.139921,0.00664,-0.102953,-0.377133,0.599468,0.078762,0.083769,-0.004268,-1.13338,-0.189282,normal.

1.455558,0.164711,1.913598,-0.436313,0.333848,-0.910736,1.105287,0.425194,-0.511913,-0.138355,-1.104573,0.757206,-0.556793,0.147843,1.870683,-2.993601,-0.35323,-0.01021,-0.667666,-0.39778,0.496736,normal.

2.203972,0.494688,-0.396079,0.035321,-0.052793,-0.40863,-0.276782,-0.013484,0.090784,-0.003902,-0.037304,0.115454,-0.030459,-0.036531,-0.3667,0.567402,0.060349,0.089906,-0.020857,-0.980287,-0.223328,normal.

2.203562,0.507587,-0.444431,0.042802,-0.055176,-0.344997,-0.312303,-0.015103,0.094417,-0.004347,-0.038894,0.091536,-0.040875,-0.04027,-0.357294,0.539828,0.053696,0.088296,-0.032804,-0.886758,-0.218746,normal.

2.212946,0.514124,-0.460363,0.045909,-0.060368,-0.336047,-0.360284,-0.010344,0.091244,-0.006055,-0.047495,0.083925,-0.058783,-0.039813,-0.370786,0.518651,0.050409,0.075349,-0.072731,-0.812923,-0.183582,normal.

2.215034,0.524585,-0.48941,0.049925,-0.060701,-0.280318,-0.389617,-0.011996,0.094656,-0.005945,-0.044563,0.062918,-0.063398,-0.042866,-0.365622,0.494363,0.045302,0.071662,-0.084455,-0.73561,-0.169578,normal.

2.219502,0.532571,-0.499207,0.050235,-0.058978,-0.232132,-0.412704,-0.012804,0.091811,-0.005195,-0.036876,0.045751,-0.062502,-0.03913,-0.366032,0.474085,0.041276,0.06585,-0.096269,-0.674957,-0.146228,normal.

3.138897,0.146795,1.405547,-0.323804,0.447077,-2.587073,1.181,0.370076,-0.324496,-0.055697,-0.516213,0.831119,0.086327,0.007047,1.227362,-1.812099,-0.205574,0.20387,0.768912,-2.43844,0.30677,normal.

2.447544,0.42909,-0.367201,0.018448,0.041522,-0.743249,0.032013,0.009921,0.115578,0.001184,0.00128,0.170109,0.044878,-0.080616,-0.374246,0.817552,0.099067,0.087672,0.048826,-0.65247,-0.259397,normal.

2.478794,0.425751,-0.314752,0.003073,0.065413,-0.705985,0.093626,0.00817,0.076384,0.007967,0.046042,0.169909,0.096378,-0.033009,-0.375375,0.845511,0.099575,0.096506,0.10126,-0.719816,-0.259237,normal.

2.527955,0.417345,-0.311119,0.013339,-0.394683,-0.753734,0.273307,0.022821,0.077213,0.011212,0.057211,0.179213,0.132984,-0.035046,-0.351556,0.895295,0.101847,0.099279,0.146685,-0.60597,-0.254225,normal.

2.482903,0.441292,-0.412305,0.007315,0.109669,-0.47226,0.234625,-0.014605,0.066015,0.019079,0.117366,0.09693,0.171603,-0.010854,-0.281746,0.864351,0.088968,0.140161,0.24882,-0.535578,-0.368373,normal.

2.510636,0.459606,-0.523473,0.042124,0.034293,-0.673133,-0.090471,0.025973,0.077078,-0.002803,-0.026196,0.140323,-0.007557,-0.047232,-0.386266,0.779358,0.081298,0.078829,-0.016002,-0.405147,-0.248698,normal.

2.487724,0.47718,-0.522037,0.031308,0.068553,-0.419951,0.004871,-0.00041,0.060769,0.009879,0.058993,0.07482,0.080817,-0.012291,-0.320414,0.762128,0.070844,0.110652,0.111731,-0.395309,-0.294045,normal.

2.658035,0.406612,-0.143535,-0.031587,0.110104,-0.849298,0.1266,0.017623,0.103814,0.014104,0.099387,0.204708,0.168839,-0.056321,-0.447167,0.915416,0.120303,0.069962,0.087954,-0.834117,-0.133957,normal.

2.624076,0.456042,-0.510389,0.033634,0.064607,-0.715209,-0.044827,0.031806,0.062805,0.001772,0.002814,0.155888,0.035102,-0.027758,-0.400058,0.839639,0.086326,0.090517,0.027788,-0.464863,-0.263039,normal.

2.596712,0.478664,-0.547316,0.030856,0.09474,-0.447877,0.037874,0.001457,0.074692,0.012944,0.077898,0.080293,0.113212,-0.022248,-0.319589,0.812508,0.074922,0.126712,0.156517,-0.420554,-0.326872,normal.

2.620422,0.482446,-0.604195,0.04644,0.063318,-0.567188,-0.103242,0.02855,0.051194,0.002376,0.008505,0.106203,0.022754,-0.014347,-0.378913,0.787225,0.071178,0.090717,0.016961,-0.285795,-0.26003,normal.

2.584145,0.517841,-0.613306,0.039746,0.083343,-0.245852,-0.086007,-0.005,0.059202,0.014246,0.089635,0.024612,0.102896,0.00158,-0.312677,0.718526,0.054511,0.123578,0.131515,-0.350058,-0.290529,normal.

2.729264,0.511276,-0.629218,0.036214,0.127058,-0.538181,-0.013093,0.100831,-0.089145,-0.015553,-0.160995,0.12777,-0.056366,0.047287,0.146767,-0.038053,-0.063135,0.177221,0.230142,-0.324824,-0.311386,normal.

2.605303,0.530168,-0.648267,0.047216,0.069506,-0.24736,-0.194383,0.007812,0.049175,0.009658,0.064422,0.014709,0.057508,0.004408,-0.34698,0.676145,0.048036,0.094089,0.03742,-0.196057,-0.214339,normal.

2.582827,0.547881,-0.646635,0.037285,0.103743,0.004497,-0.099252,-0.021794,0.055054,0.022019,0.148764,-0.054207,0.146952,0.016919,-0.276225,0.656253,0.039296,0.126085,0.166306,-0.184589,-0.259398,normal.

2.616567,0.547101,-0.693341,0.053917,0.063991,-0.182049,-0.271597,0.012455,0.038841,0.008218,0.059157,-0.012283,0.034475,0.0125,-0.357645,0.631948,0.038825,0.077372,-0.014781,-0.045687,-0.165307,normal.

2.60383,0.560159,-0.681666,0.044442,0.089914,0.003458,-0.207555,-0.007843,0.037187,0.017545,0.123851,-0.062905,0.101186,0.027672,-0.31226,0.614812,0.032215,0.096534,0.071804,-0.031164,-0.18331,normal.

2.924699,0.43783,-0.135904,-0.047482,0.169616,-0.757564,0.075596,0.020551,0.110354,0.02511,0.181822,0.175181,0.259548,-0.044395,-0.485625,0.939484,0.11706,0.074044,0.12596,-0.866127,-0.062611,normal.

2.841462,0.513797,-0.597938,0.024166,0.154707,-0.313632,0.022581,-0.002471,0.081633,0.024711,0.159132,0.042544,0.207843,-0.008022,-0.321893,0.845516,0.068114,0.155588,0.247417,-0.468406,-0.334339,normal.

2.801038,0.537385,-0.64901,0.023908,0.180394,-0.041763,0.099745,-0.030558,0.07699,0.035218,0.228012,-0.031574,0.276858,0.013294,-0.241089,0.815297,0.054224,0.192828,0.372552,-0.415942,-0.406931,normal.

2.944832,0.457386,-0.399881,-0.003936,0.162542,-0.717947,0.05442,0.03079,0.092938,0.018792,0.128413,0.147707,0.192076,-0.038467,-0.435726,0.949498,0.101859,0.098478,0.128815,-0.538239,-0.192403,normal.

3.172645,0.243151,1.15222,-0.27639,0.338816,-1.463945,0.622156,0.011862,0.236298,0.060096,0.478392,0.378127,0.666012,-0.153617,-0.740851,1.251196,0.247688,-0.067662,0.1626,-1.758714,0.544661,normal.

2.911141,0.472946,-0.586009,0.030183,0.165322,-0.60723,0.099712,0.004753,0.21899,0.016187,0.110736,0.082159,0.178097,-0.169484,-0.330914,0.933864,0.099206,0.130887,0.192677,-0.275029,-0.317269,normal.

EK 2. Genetik Algoritma kullanılarak detektör üretimi aşağıdaki gibidir.

Rulet çarkı yöntemiyle ebeveynler seçildikten sonra çaprazlama işlemi aşağıdaki gibidir. Çaprazlama sırasında hangi genden sonra işlem yapılacağı k değeri ile ifade edilmiştir.

k = 2 olmak üzere ;

1.Ebevyn Kromozom : 1.14038 0.457781 **|** 0.597827 0.064336

2.Ebevyn Kromozom : 3.189038 0.498095 **|** 8.203319 30.6949

1.Çocuk Kromozom : 1.14038 0.457781 | 8.203319 30.6949

2.Çocuk Kromozom : 3.189038 0.498095 | 0.597827 0.064336

Mutasyon işlemi aşağıdaki şekilde yapılmaktadır.

k = 3 ve r = 0.1 olmak üzere;

1.Çocuk Kromozom : 1.14038 0.457781 8.303319 30.6949

2.Çocuk Kromozom : 3.189038 0.498095 0.697827 0.064336

Yukarıdaki işlemde k değiştirilecek genin sırasını r ise mutasyon sırasında kullanılacak değeri göstermektedir. Mutasyon işlemi sonrası uygunluk değeri hesaplanarak yeni bireylerin detektör olup olmayacaklarına karar verilmektedir.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Feyzan SARUHAN ÖZDAĞ
Doğum Yeri	Kızıltepe
Doğum Tarihi	28.01.1991
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	05063083679
E-Posta Adresi	feyzansaruhan@gmail.com
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	06.07.2012

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği
Programı	Bilgisayar Mühendisliği
Mezuniyet Tarihi	22.06.2017