



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**GRİ KURT OPTİMİZASYON (GKO)
ALGORİTMASI VE YAPAY SİNİR AĞI (YSA)
KULLANILARAK HİBRİT BULUT TABANLI
SALDIRI TESPİT VE YANIT SİSTEMİ**

Ismail Mohamed Nur

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ocak-2021
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ismail Mohamed Nur tarafından hazırlanan “ Gri Kurt Optimizasyon (GKO) Algoritması ve Yapay Sinir Ağı (YSA) Kullanılarak Hibrit Bulut Tabanlı Saldırı Tespit ve Yanıt Sistemi ” adlı tez çalışması 29/01/2021 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof.Dr. Erkan ÜLKER

Danışman

Prof.Dr. Erkan ÜLKER

Üye

Dr.Öğr.Üyesi Alper KILIÇ

Üye

Dr.Öğr.Üyesi Özkan İNİK

İmza

.....

.....

.....

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ismail Mohamed Nur

Tarih: 29/01/2021

ÖZET

YÜKSEK LİSANS TEZİ

GRİ KURT OPTİMİZASYON (GKO) ALGORİTMASI VE YAPAY SİNİR AĞI (YSA) KULLANILARAK HİBRİT BULUT TABANLI SALDIRI TESPİT VE YANIT SİSTEMİ

Ismail Mohamed Nur

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Prof.Dr. Erkan ÜLKER

2021, 87 Sayfa

Jüri

**Prof.Dr. Erkan ÜLKER
Dr.Öğr.Üyesi Alper KILIÇ
Dr.Öğr.Üyesi Özkan İNİK**

Teknoloji hızla büyümekte ve bulut bilişim kullanımı da aynı hızla artmaktadır. Bugünlerde irili ve ufaklı şirketlerin çoğu bulutu kullanmaktadır. Bulut bilişim, kullandıkça öde mantığıyla çalıştığı için ekonomik fayda sağlamaktadır. Bulut kullanımının artmasıyla birlikte buluttaki güvenlik sorunları da artmıştır. Bu sorunların çözümü için, güvenlik duvarı, güvenlik açığı tarayıcıları ve saldırı tespit sistemi (IDS) gibi bazı mekanizmalar ve saldırıları azaltmak için diğer yöntemler kullanılmıştır. Ancak yeni ve önceden bilinmedik şekilde oluşan saldırıların olması durumunda bu tarz sistemler buluta karşı yapılan saldırıları tespit etmek için yetersiz kalmaktadır. Bulut güvenliğini tehditlerden ve zayıf noktalardan korumak için çeşitli güvenlik yöntemleri vardır. Bu tez çalışmasında, bulut üzerindeki yapılan saldırıları tespit etmek ve sistem güvenliğini sağlamak için Gri Kurt Optimizasyon (GKO) ve Yapay Sinir Ağı (YSA) dayalı yeni bir melez/hibrit bulut tabanlı Saldırı Tespit Sistemi (IDS) önerilmiştir. GKO, güvenlik, tıp / sağlık, optimizasyon, mühendislik ve bilgi işlem gibi birçok alanda etkili olarak kullanılan meta-sezgisel algoritmalarından biridir. Tezde, YSA eğitmek için GKO kullanılmıştır ve sonuçlar diğer sınıflandırma algoritmalarıyla karşılaştırılmıştır. Deneysel çalışmalarda, UNSW-NB15 ve NSL-KDD gibi mevcut saldırı tespit veri setleri kullanılmıştır. Simülasyon sonuçları, önerilen algoritmanın saldırı tespitinin doğruluğunu NSL-KDD ve UNSW-NB15 veri kümeleri için sırasıyla % 97.83 ve % 98.21 artırdığını göstermiştir.

Anahtar Kelimeler: Bulut Bilişim, Saldırı tespit sistemi, Gri Kurt Optimizasyon Algoritması, Yapay Sinir Ağı, Siber Güvenlik.

ABSTRACT

MS THESIS

A HYRID CLOUD-BASED INTRUSION DETECTION AND RESPONSE SYSTEM USING GREY WOLF OPTIMIZATION (GWO) ALGORITHM AND ARTIFICIAL NEURAL NETWORK (ANN)

Ismail Mohamed Nur

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

Advisor: Prof.Dr. Erkan ÜLKER

2021, 87 Pages

**Jury
Prof.Dr. Erkan ÜLKER
Assist.Prof.Dr. Alper KILIÇ
Assist.Prof.Dr. Özkan İNİK**

Technology is growing rapidly and the use of cloud computing is increasing at the same speed. Most large and small companies use the cloud these days. Cloud computing provides economic benefits because it works with a pay-as-you-go logic. Security problems in the cloud have increased with the increase in cloud usage. To solve these problems, some mechanisms such as firewall, vulnerability scanners and intrusion detection system (IDS) and other methods to mitigate attacks are used. However, in case of new and previously unknown attacks, such systems are insufficient to detect attacks against the cloud. There are various security methods to protect cloud security from threats and vulnerabilities. In this thesis, a new hybrid cloud-based Intrusion Detection System (IDS) based on Gray Wolf optimization (GWO) and Artificial Neural Network (NN) has been proposed in order to detect attacks and ensure system security on the cloud. GWO is one of the meta-heuristic algorithms used effectively in many fields such as security, medicine / health, optimization, engineering and computing. In the thesis, GWO was used to train the Artificial Neural Network (NN) and the results were compared with other classification algorithms. In experimental studies, the current intrusion detection data sets such as UNSW-NB15 and NSL-KDD were used. The simulation results showed that the proposed algorithm increased the accuracy of intrusion detection by 97.83% and 98.21% for the NSL-KDD and UNSW-NB15 datasets, respectively.

Keywords: Cloud Computing, Intrusion Detection System, Grey Wolf Optimizer, Artificial Neural Network, Cyber Security.

ÖNSÖZ

Bu tez çalışmasını bitirmek için iyi bir çaba harcandı. Bununla birlikte, bu çalışma birçok kişinin yardımı, dostluğu ve desteği olmadan yapılamazdı. Hepsine içten teşekkür etmek benim için bir onurdur. Ayrıca, danışmanım **Prof.Dr. Erkan ÜLKER**'e; rehberliği ve sürekli kılavuzluğundan dolayı, bana bu kadar dikkat ve zaman harcaması ve tezin tamamlanmasındaki desteği, tez için gerekli tüm temel bilgileri sağladığı için teşekkür ediyorum.

Bu tez çalışmasını tamamlarken yanımda olan sevgili ailem (Annem, Babam ve kardeşlerim) ve arkadaşlarıma gösterdikleri işbirliği ve teşvik için çok teşekkür ederim.

Ismail Mohamed Nur
KONYA-2021

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
1.1. Tezin Amacı.....	1
1.2. Tezin Önemi	3
2. KAYNAK ARAŞTIRMASI	5
3. MATERYAL ve METOT	13
3.1. Bulut Bilişim.....	13
3.1.1. Bulut bilişimin sağladığı temel özellikler ve imkanlar	14
3.1.2. Bulut bilişim mimarisi	16
3.1.3. Bulut bilişim servis modelleri	18
3.1.4. Bulut bilişim dağıtım modelleri	20
3.1.5. Bulut bilişim avantajları.....	21
3.1.6. Bulut bilişim dezavantajları	24
3.2. Bilgi Güvenliğine Yönelik Tehditler ve Saldırı Tespiti.....	26
3.2.1. Bilgi güvenliğine yönelik tehditler	26
3.2.2. Saldırı tespiti	30
3.2.3. Bulut bilişime yapılan saldırılar.....	43
3.2.4. Sistem bileşenleri	45
3.3. Makine Öğrenmesi.....	46
3.3.1. Gözetimli öğrenme	46
3.3.2. Gözetimsiz öğrenme	48
3.3.3. Pekiştirmeli öğrenme	49
3.4. Optimizasyon ve Meta-sezgisel yaklaşımlar	50
3.5. Hibrit / Melez algoritmanın (GWO-NN) alt yapısı	51
3.5.1. Sinir ağları (Neural networks)	52
3.5.2. Gri kurt optimizasyonu (Grey wolf optimization GWO)	55
3.6. Kullanılan Veri Setleri	60
3.7. CloudSim	64
3.7.1. Veri merkezi	66
3.7.2. Veri merkez brokeri	66
3.7.3. Host.....	66
3.7.4. Sanal makine.....	67
3.7.5. Cloudlet.....	67
4. ÖNERİLEN GWO VE NN HİBRİT ALGORİTMASI.....	68

4.1 Deneyisel Çalışmalar	70
5. SONUÇ	80
KAYNAKLAR	82
ÖZGEÇMİŞ	87



SİMGELER VE KISALTMALAR

Kısaltmalar

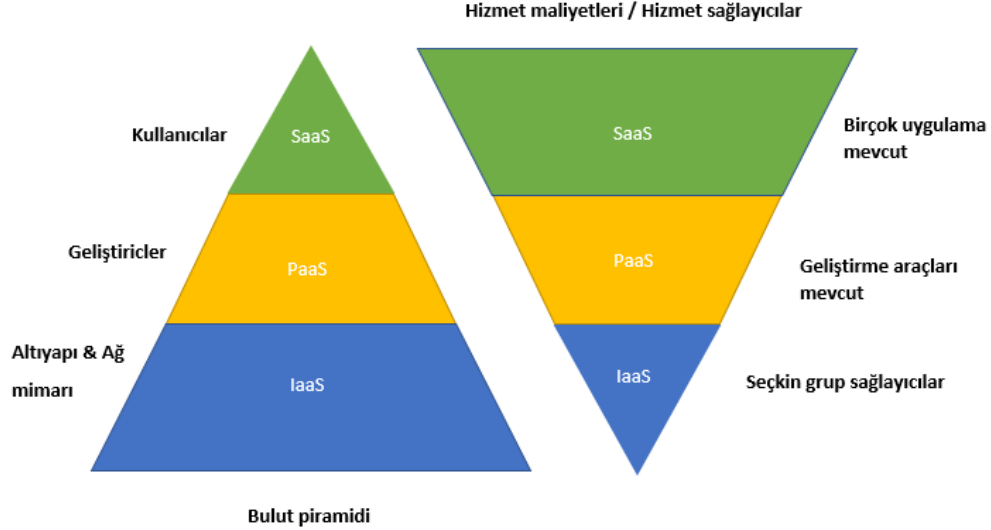
CC	Bulut Bilişim (Cloud Computing)
CIA	Gizlilik, Bütünlük, Kullanılabilirlik (Confidentiality, Integrity, Availability)
CPU	Merkezi İşlem Birimi (Central Processing Unit)
CSP	Bulut Servis Sağlayıcısı (Cloud Service Provider)
DDoS	Dağıtılmış Hizmet Reddi (Distributed Denial of Service)
DIDS	Dağıtılmış Saldırı Tespit Sistemi (Distributed Intrusion Detection System)
DKSM	Doğrudan Çekirdek Yapısı Manipülasyonu (Direct Kernel Structure Manipulation)
DNS	Alan Adı Sunucusu (Domain Name Server)
DoS	Hizmet Reddi (Denial of Service)
GIDP	Genelleştirilmiş Daldırı Tespiti ve Önleme (Generalized Intrusion Detection & Prevention)
GWO	Gri Kurt Optimizasyon (Grey Wolf Optimization)
HIDS	Host tabanlı Saldırı Tespit Sistemi (Host-based Intrusion Detection System)
IaaS	Hizmet Olarak Altyapı (Infrastructure as a Service)
IT	Bilgi Teknolojisi (Information Technology)
IDAR	Saldırı Tespiti ve Uyarlanabilir Yanıt mekanizması (Intrusion Detection & Adaptive Response mechanism)
IDMEF	Tespit Mesajı Değişim Biçimi (Intrusion Detection Message Exchange Format)
IDS	Saldırı Tespit Sistemi (Intrusion Detection System)
IDRS	Saldırı Tespit ve Yanıt Sistemi (Intrusion Detection and Response System)
IPS	Saldırı Önleme Sistemi (Intrusion Prevention System)
IRS	Saldırı Yanıt Sistemi (Intrusion Response System)
MCC	Mobil Bulut Bilişim (Mobile Cloud Computing)
NIDS	Ağ tabanlı Saldırı Tespit Sistemi (Network-based Intrusion Detection System)
OS	İşletim Sistemi (Operating System)
PaaS	Hizmet olarak Platform (Platform as a Service)
SaaS	Hizmet olarak Yazılım (Software as a Service)
SMD	Akıllı Mobil Cihazlar (Smart Mobile Devices)
VM	Sanal Makine (Virtual Machine)
XaaS	Hizmet olarak Her Şey (Anything as a Service)

1. GİRİŞ

1.1. Tezin Amacı

Bilgi çağının belirlediği modern çağın getirdiği yenilikler ve hızla değişen kullanıcı ihtiyaçları ile bilişim teknolojileri değişim ve gelişmeyi sürdürmektedir. Günümüzde, dinamik ofis yapısı nedeniyle masa başı kullanıcı sayısındaki artış kurumlarda daha düşüktür. Kurumların daha az kaynak ve ekonomik nedenlerle daha fazla hizmet sunmaları beklenmektedir, bu yüzden kaynakların verimli kullanımı düşüncesi kendilerini yeniden yapılandırmak için önemli bir hal almıştır. Bilgi sektöründe ortaya çıkan bu ihtiyaçlar için, fiziksel sistemleri sanal ortamlara taşımak ve birçok sanal sistemi fiziksel bir sistemde kullanmak için çözümler geliştirilmiştir. Zaman içinde, "bulut bilişim" kavramı, sanallaştırma geliştirilmesinden ve hızlı bir şekilde benimsenmesinden sonra, kullanıcıların alan, zaman ve platformdan bağımsız olarak uygulamalarını kullanma yönünde artan bir talep doğrultusunda ortaya çıkmıştır. Sanallaştırma altyapısında, yeni bir bilgi teknolojisi olarak bulut bilişim yapılandırılmıştır.

Bulut bilişim, Bilgi Teknolojileri'nde (IT) sistematik olarak kullanılan en yeni yaklaşımlardan biridir. Bulut bilişim genellikle kullandığın kadar öde modeline ve yaklaşımına sahiptir. Herhangi bir bulut kullanıcısı, hizmet erişim oranına göre ödeme yapmaktadır. Bulut bilişimde farklı servis ve dağıtım modelleri vardır. Bulut hizmetlerinin maliyeti ve modeller arasındaki ölçek Şekil 1.1'de gösterilmektedir. Ayrıca, bulut, Bilişim Teknolojileri (IT) sektörü için en güçlü/etkili dağıtım bileşeni ve en çok tercih edilen servis haline gelme yolunda ilerlemektedir.



Şekil 1.1. Bulut piramidi ve Servis maliyetleri

Bulut bilişim üzerinden sağlanan hizmetler, Hizmet Olarak Her Şey (Anything as a Service - XaaS) olarak tanımlanmaktadır. Bulut bilişim hizmetlerinin üç temel bileşeni vardır:

- 1) Hizmet Olarak Yazılım (Software as a Service - SaaS); Son kullanıcılar için geliştirilen ve web’de kullanılan bir hizmettir(örneğin, Salesforce.com, Google’ın posta servisi veya Gmail v.s. gibi).
- 2) Hizmet Olarak Platform (Platform as a Service - PaaS): Programlama / kodlama uygulamaları yapmak ve bu uygulamaları verimli ve hızlı bir şekilde dağıtmak için uygulanan bir araç ve hizmet grubudur (örneğin, Salesforce1 ve Google App Engine gibi).
- 3) Hizmet Olarak Altyapı (Infrastructure as a Service - IaaS): Tüm buluta güç sağlayan ana hizmettir. Yani işletim sistemleri, depolama, sunucular ve ağlar gibi tüm buluta güç sağlayan donanım ve yazılımı içerir. (örneğin. Amazon EC2 ve Windows Azure ve Rackspace) (Mell & Grance, 2011).

Kullanıcılar Hizmet olarak Altyapı'yı (IaaS) kullanmayı seçtiğinde, sağlayıcı bir hizmet olarak donanım ve erişim arayüzleri sunar. Altyapı için, kullanıcılar işletim sistemi (OS), yazılım veya herhangi bir şebeke servisi için lisans ücreti ödememektedir. IaaS özel donanım (ağ, depolama, CPU, sunucu vb.) ve sanallaştırılmış medya sağlayıcıları tarafından desteklenir. Hizmet Olarak Platform (PaaS), ara katman yazılımı ve çalışma zamanı sağlayıcıları tarafından çalıştırılan IaaS'ye eklenir. PaaS'a Hizmet

Olarak Yazılım (SaaS) eklenir, veri ve ağ hizmetleri sağlayıcılar tarafından sağlanmaktadır. Tedarikçi tarafından yönetilen parçalar arttıkça maliyetler artmaktadır.

Bu modellerde, güvenlik gizliliği sağlamak için temel gereksinimlerden biridir ve proaktif bir savunma mekanizması oluşturmanın temel yolu saldırı tespit yada önlemedir. Bulut bilişim, özellikle kullanıcılara sağlanan ekonomik fayda için birçok fayda sağlar. Bu avantajların yanı sıra, bazı zorluklar da içermektedir. Güvenlik, geleneksel bilişim teknolojilerinde olduğu gibi bulut bilişimdeki en büyük sorunlardan biridir. Güvenlik, bulut bilişimde çözülmesi gereken en önemli zorluktur. Bulut bilişimde güvenlik sorunu kritik bir sorudur. Bu sorun nedeniyle, kullanıcılar bulutları kullanmakta tereddüt etmektedirler. Sistem yöneticileri, geleneksel bilgi merkezlerinden bulut bilgi işleme geçişi için ödeme yapmak zorundadır. Virüsten koruma yazılımı, güvenlik duvarları, güvenlik sistemleri ve izinsiz giriş tespit / önleme sistemleri gibi mekanizmaları kullanarak güvenlik sorunlarının üstesinden gelmek için çaba harcanmaktadır. Bu mekanizmalara karşı performans açısından en büyük engel, özellikle Merkezi İşlem Birimi (CPU) ve bellek (Memory) gibi ağır donanım kullanımıdır.

Günümüzde, şirketlerin çoğu bulut tabanlıdır ve bulut saldırıları her geçen gün artmaktadır. Bu yüzden, buluttaki güvenlik çok kritik bir konudur. Buluta yapılan saldırıyı azaltmak için birçok teknik geliştirilmiştir, ancak bu sistemler tek başına yeterli değildir çünkü en son veya yeni farklı saldırıları tespit edemezler. Bu nedenle, gelişmiş / yeni sistemler geliştirilmelidir. Birçok araştırmacı çözüm bulmak için bu konuda çalışmaktadır.

Bu tezde, bulut tabanlı saldırı tespit ve yanıt sistemleri incelenmiştir. Yeni ve bilinmedik saldırıları tespit etmek için makine öğrenme algoritmaları kullanılmıştır. Bu, otomatik önleme özellikli akıllı bulut tabanlı bir sistemin geliştirilmesine ve uygulanmasına yardımcı olmuştur.

1.2. Tezin Önemi

Sistem bulut üzerindeki saldırıları tespit edebilecek ve bulabilecek herhangi bir mevcut saldırı sorununu çözebilecek yetkinliğe kavuşturulmuştur. Ayrıca, ağdaki yeni müdahaleleri tespit etmek için sistemde makine öğrenme teknikleri kullanılmıştır. Bulut kullanıcısına / tüketicisine ve bulut sağlayıcısına buluttan izinsiz giriş paketleri alıp almadıklarını bildirmek için kullanılan yöntemlerle bunun bir saldırı olduğu/olmadığı algısı üretilerek saldırı algılanan doğrulukla önlenecektir. Aynı zamanda bulut tabanlı

řirketlerin ve arařtırmacıların kötü niyetli saldırıları önceden tespit etmesine ve önlenmesine yardımcı olacaktır. Sistemin yeni saldırıları çözecek kadar iyi olduğuna inanmaktayız. Son olarak, insan bilgisi kısıtlaması nedeniyle güvenlik için mükemmel bir sistemin uygulama olarak üretilmesi garanti edilemeyecektir, ancak yeni yaklaşımlar, veriler ve önceki saldırıların deneyimini kullanarak daha iyi güvenlik sistemleri uygulanabilecektir.



2. KAYNAK ARAŞTIRMASI

Kötü amaçlı yazılımlar, ortalama (phishing), DDoS, fidye yazılım (ransomware) ve diğer saldırı türleri günlük olarak karşımıza çıkmakta olan tehlikelerdir. Bir işletme, her 40 saniyede veya belki daha az sürede fidye yazılım saldırısından etkilenmektedir. Siber tehditler, bulut tarafından desteklenen her şeyi etkilemiştir. Siber saldırıların yol açtığı hasarın, 2021 yılına kadar yılda 6 trilyon doları bulması beklenmektedir ("Cisco Global Cloud Index," 2018). Siber saldırılar; yanlış veya kötüye kullanım, verimlilik kaybı, parasal kayıplar, finansal ve kişisel verilerin çalınması, fikri mülkiyet hırsızlığı, verilerin imha edilmesi ve zarar görmesi, verilerin silinmesi, sistemlerin hacklanması gibi durumları kapsamaktadır. 2017 White Hat'ın uygulama güvenliği raporuna göre, 2016'da bildirilen ihlallerin %30'u bulut/web uygulamalarına yönelik saldırılarla ilgilidir ("Cisco Global Cloud Index," 2018). 2017 yılında çok sayıda siber saldırı gerçekleşmiştir. CIA Vault 7 korsanlığı, wanna-cry fidye yazılımı ve equifax veri ihlali mevcut güvenlik açıkları örnek olarak gösterilmiştir. Cisco nun yaptığı 2017 çalışmasında, bir saldırı geçirmiş olan şirketlerin yaklaşık dörtte birinin iş fırsatlarını kaybettiği gözlemlenmiştir ("Cisco Global Cloud Index," 2018). Her beş şirketten biri, bir saldırı sonucu müşterilerini kaybetmiş ve yaklaşık %30'luk bir gelirden olmuştur. Örneğin, Equifax'ın kredi verilerinin ihlali neticesinde, çok sayıda kişisel ismin, ehliyet numarasının, sosyal güvenlik numarasının ve kredi kartı bilgilerinin yayınlanmasıyla sonuçlanmıştır, bu nedenle şirket ve müşterileri için ciddi kayıplar ve bazı önemli talihsizlikler görülmüştür ("Cisco Global Cloud Index," 2018).

Bulut bilişimin birçok faydasına rağmen, birçok şirket güvenlik riskleri ve zorlukları nedeniyle bilgilerini buluta aktarma konusunda hala kararsızdır. Bilgi sistemleri güvenliği için güvenlik duvarları, güvenlik açığı tarayıcıları ve saldırı tespit sistemleri kullanılmaktadır. Bu güvenlik mekanizmalarından herhangi birinin tek başına kullanılması, güvenlik açısından yeterli değildir. Çünkü her biri farklı açılardan güvenlik konularına odaklanmıştır (NUR & ÜLKER, 2018). Sistemde güvenliğin sağlanması, bu mekanizmaların birbirlerini desteklemek için birlikte kullanılmasını gerektirir. Ayrıca, bulut bilişim sistemleri güvenliği sağlamak için çeşitli yönetim modellerine odaklanmıştır. Bu modeller temel olarak saldırı tespit sistemi, veri şifreleme ve güvenli bilgi işlem gibi teknikleri içerir (Kumar, Kumar, Singh, & Kumar, 2012). Kuruluşlar ve araştırmacılar, makine öğrenmesi gibi teknolojilerle izinsiz giriş tespit etme, bunlara

müdahale etme ve tehdit verilerini daha etkin / verimli bir şekilde analiz etme becerisini geliştirebilecek yeni yollar keşfetmektedir.

Alharkan ve ark., çoklu melez makine öğrenme yöntemlerinin oluşturulmasını belirten yeni bir çalışma önermiştir. Amaçları, siber güvenlik ağında izinsiz giriş tespit sisteminin performans kapasitesini değerlendirmek için bir sistem sunmaktır. Deneysel sonuçları, danışmanlı ve danışmansız yöntemlerin birbirini tamamlayan, bilinen ve bilinmeyen kötü niyetli faaliyetler için potansiyel tanımlama ve yeterli tespiti sahip yeni bir model üretme görevine doğru büyük bir entegrasyon ilerlemesi olduğunu göstermektedir (Alharkan & Martin, 2012).

2012 yılında Li ve ark. (Li, Sun, & Wang, 2012) sinir ağı temelli saldırı tespit sistemi olan bir yöntem sunmuşlardır. Yeni model son derece adaptif mimariye dayalı dağıtık bir sistemdir. Bu, buluttaki herhangi bir makinenin aşırı yüklenmesine mücadele etmeden, tam bir kaynak kullanılabilirliğini sağlamıştır. Ayrıca, sinir ağlarına dayalı makine öğrenme algoritmaları, yeni modelin kanıtlanmış ve optimum doğruluk kazanımlarıyla ileri düzey saldırıları durdurabilmesini sağlamıştır. Yöntemin performans değerlendirmeleri neticesinde, bulut altyapısındaki herhangi bir saldırı olayını tespit etmek için yeterli potansiyele sahip olduğu gösterilmiştir. KDD veri seti kullanılarak, modelin, test değerlendirme sürecinde % 99 doğruluk elde edilmiştir.

Kholidy ve Baiardi, bulut sistemleri için maskeli saldırılar hakkında çalışma yapmışlardır, saldırganın bulut kaynaklarını denetlemesine olanak sağlandığını öğrenmişlerdir (Kholidy & Baiardi, 2012a). Mevcut veri kümelerinin bulut kullanıcılarının tipik davranışlarını ihmal ettikleri için, bulut IDS'lerini eğitmek ve değerlendirmek için bu veri kümelerinin uygun olmadığını belirtmişlerdir. CIDD isimli bir bulut saldırı tespiti veri seti önermişler ve bu veri seti ile çok sayıda saldırının tespit edilebileceğini söylemişlerdir. CIDD'yi oluşturmak için, ana bilgisayar tabanlı günlük dosyalarını ve ağ paketlerini ayrıştırma ve analiz etmek için bir log analizörü ve korelatör sistemi geliştirmişler. CIDD'nin daha fazla arıtılması için CIDD denetimlerini gerçek bir bulutta uygulamayı ve maskeli saldırıları tespit etmek için (Kholidy & Baiardi, 2012b) HSGAA yaklaşımlarını kullanmayı planlamışlardır.

Genetik algoritma (GA) ve bulanık destek vektör makinesi (SVM) sınıflandırmasına dayanan bir izinsiz giriş tespit modeli, 2012 yılında Kannan ve ark. (Kannan, Maguire Jr, Sharma, & Schoo, 2012) tarafından sunulmuştur. GA, özellik seçimi için kullanılırken bulanık SVM, GA ile azaltılmış özellikleri sınıflandırmak için kullanılmıştır. Özellik seçimi, modelleme sınıfındaki modeli yavaşlatan önemsiz

özellikleri çıkarmak için yapılmıştır. Buna ek olarak, azaltma özelliği modeli boyutlandırırken tatmin edici doğrulukta sonuçlar elde etmek için kullanılmıştır. Yeni sistemin testi için KDD kullanılmış, yüksek algılama oranı ve daha az yanlış alarm oranı ile anormallik tehditleri tespit edilmiş ve FSVM algoritması ile %98.51 doğruluk elde etmişlerdir.

Özellikle Dağıtık Hizmet Reddi (DDoS) saldırısı tespitine odaklanan bir bulut saldırı tespit sistemi, Lonea ve ark. tarafından önerilmiştir. Bulut ortamındaki DDoS saldırılarının çoğu sunucuları çökertmek amacıyla bulut sunuculardan gelmektedir. Bu problemin çözümü, bulut sistemlerin sanal makinelerinde (VM'ler) yer alan saldırı tespit sistemlerinden alınan verilerle, daraltılmış bir ön uça birleştirme yapmaktan geçmiştir. İstenmeyen bir olay gerçekleştiğinde, VM'lerin saldırısı alarm verecek bir eylemde bulunacaktır. Bu şüpheli eylem alarmı, ön uç sunucunun bulut birleştirme birimindeki (cloud fusion unit CFU) MYSQL veritabanına kaydedilecektir. Dempster-Shafer teorisi (DST) ve hata ağacı analizi (fault-tree analysis FTA) kullanılarak alarmın verildiği bir dizi çözüm sunmuşlardır (Lonea, Popescu, & Tianfield, 2013).

Günümüzde, birçok ağ ve veri merkezi, uygun sunucuları güçlendirmek için bulut bilişimi önemli ve gerekli bir yöntem olarak uygulamaya başlamıştır. Bulut bilişim çeşitli avantajlara sahip olmasına rağmen, güvenlik en çok vurgulananlardan biridir. Askerlik hizmetleri, banka transfer geçişleri ve özel hükümet faaliyetleri, dürüstlük ve erişilebilirlik vs gibi diğer güvenlik temelleri arasında en önemli olarak Gizlilik gereksinimlerine ihtiyaç duyulmaktadır. Oktay ve ark, bulut kullanıcılarının yetkisiz kullanıcıların ve kötü niyetli bulut sağlayıcılarının politikalarını tespit etmek için bulut bilişim için hibrit saldırı tespitini sunmuşlardır. Yeni geliştirilmiş modellerini AdjointVM olarak isimlendirmişlerdir (Ucman Oktay, Aydın, & Sahingoz, 2013). Araştırmacıların amacı, kullanıcılar için güvenli bir bulut platformu geliştirmek ve Adjoint sisteminin değişkenlik bölümlerinden birini başarmaktır. Burada, tespit edilen VM ve dedektör VM olan VM'yi tespit etmek için iki VM konfigürasyonuna ihtiyaç vardır. VM'lerin şekli, saldırganları izlemek için dairesel bir zincire birleştirilir. AdjointVM'in aksine, çalışan VM sayısı ve algılanması gereken diğer VM sayısı orantılıdır.

Bulut bilişim, buluta tek başına erişim için minimum maliyet olan kullanım başına ödemeli olarak internet üzerinden veri depolama hizmetlerini uzaktan koruyan ve önemli bir alandır. İnternet üzerinden insanlar, bulut bilişim kullanarak bilgilerini paylaşmaktadırlar; ne yazık ki bunu fırsat bilerek bulut ortamındaki zayıf noktalardan yararlanıp düzenli olarak insanların bilgilerini çalan ve bunlara müdahale eden bazı

saldırganlar mevcuttur. Genelde saldırgan, ana bilgisayar tabanlı ve ağ tabanlı bir saldırı yapıp yapılmayacağına karar vermektedir. Bu değişkenlik tarzının üstesinden gelmek için araştırmacılar Gupta ve Kumar (Gupta & Kumar, 2015), saldırganları ön plana çıkaran bir makale önermişlerdir. Makaleleri, bir bulut bilişim platformundaki hostlardaki saldırganları özellikle rootkit'leri, solucanları ve truva atlarını vurgulamaktadır. Dahası, çalışmaları, rahatsız edici sistem çağrısı tespiti için bazı yöntemler göstermektedir ve ayrıca çalışma, buluttaki kötü program uygulamalarını keşfeden yeni bir sistem çağrısı imza yapısı tabanlı yöntemi içermektedir. Önerilen yöntemlerin, ilgili komplikasyon ve kullandığı kaynaklar açısından üstünlükleri bulunmaktadır. Çalışmalarını tüm UNM (New Mexico Üniversitesi) saldırı veri setleri üzerinde denemişler ve elde ettikleri en iyi deneysel sonuç %98 doğruluk olmuştur (Gupta & Kumar, 2015).

Ling ve ark., AdjointVM'de çalışan gelişmiş bir mekanizma geliştirmiştir. Saldırganlar, bitişik iki VM'ye saldırırlarsa bulut hizmetlerinde tamamen mağduriyet yaşanabilecektir (Ling, Hsien, & Hwang, 2016). Makalelerinde VM'nin çift IDS tespitine sahip olabileceği çift dairesel zincir yöntemi kullanmışlardır. Yeni mekanizma, bitişik iki VM'nin güvenli bir şekilde algılandığını algılayabilmektedir. Mekanizmaları VM'lerin güvenliğini artırmakla birlikte zaman tüketimi fazla olmuştur.

Mehmood ve ark. bulut güvenliği sorunlarına genel bir bakış atmışlardır. Bulutta var olan bazı IDS'leri tespit süreleri, veri kaynakları, ID tipleri, konumlandırma ve tespit teknikleri ile ilgili analiz etmişlerdir. Ayrıca analiz, bulut bilişim ortamının güvenlik gereksinimlerini karşılayıp karşılamadıklarını değerlendirmek adına her teknik için sınırlamalar sağlamıştır. Buluttaki güvenlik sorunlarını gidermek için birden fazla algılama yöntemi kullanarak IDS dağıtmaya odaklanmışlardır (Mehmood, Shibli, Habiba, & Masood, 2013).

Gizlilik, bütünlük ve kullanılabilirlik her ağ tabanlı teknoloji için önemlidir. Bulut bilişim güvenliği, dağıtılmış saldırılara ve diğer kötü amaçlı saldırılara karşı önlem almak için aynı sorunlarla karşı karşıyadır. Al Haddad ve ark., ağ trafiğini izleyerek buluttaki saldırıları azaltmak için Ortak İşbirliği Ağı Tespit Sistemi (C-NIDS) olarak adlandırdıkları yeni bir güvenlik yöntemi uygulamışlardır. Bilinmeyen atakları tespit etmek için SVM'yi ve bilinen ataklar için imza bazlı hafifletme olarak Snort'u kullanmışlardır. İşbirlikçi bir şekilde çalışmak ve bulut altyapısına karşı başlatılan saldırılara direnmek için bulutta algılayıcılar kullanmışlardır. Bu sayede bilgi tabanı güncel kalabilmiştir (Al Haddad, Hanoune, & Mamouni, 2016).

Youssef ve ark., risk değerlendirme metodolojilerini kullanarak saldırı modelini analiz edip bulut ortamında siber saldırıları tespit etmek için yeni bir yaklaşım önermişlerdir. Önerilen çözümünün amacı, bir bulutta konuşlandırılmış Saldırı Tespit Sistemlerinden (IDS) elde edilen kanıtları, her saldırı modeliyle ilgili risk değerlendirmesi ile birleştirmektir. Yaklaşımda bulut ortamlarına yönelik dağıtılmış ve çok aşamalı saldırıların etkisini ve olasılığını analiz eden her belirti, gösterge ve güvenlik açığını analiz etmek için yeni bir niteleyici çözüm gösterilmiştir. Yaklaşımın uygulamasının yanlış uyarı sayısını azaltacağı ve IDS performansını artıracacağı iddia edilmiştir (Youssef, Nada, Elmehdi, & Boubker, 2016).

Bulutun sanal ağ katmanındaki (VNL) saldırıları tespit etmek için geliştirilmiş bir güvenlik platformu Arjunan ve Modi (Arjunan & Modi, 2017) tarafından uygulanmıştır. Amaçları, karar ağacı, naive Bayes, rastgele orman vb. gibi farklı sınıflandırıcıları kullanarak olası saldırıları azaltmak ve önlemek için anomali temelli ve imza yaklaşımlarını bütünleştirmektir. Her kümedeki ve buluttaki dağıtılmış saldırıları azaltmak için, bulutun her bölgesinden yapılan saldırı kanıtlarını toplar ve son karar verme için Dempster-Shafer (DST) teorisini uygular. Çevrimdışı simülasyon yoluyla bulut IDS gereksinimine sahip farklı veri setleri kullanılarak analiz ve test edilmiş olup %99 doğruluk elde edilmiştir.

Algahtani ve John tarafından (Algahtani & John, 2017) Naive Bayes, KNN, J48 ve OneR gibi en etkili sınıflandırma algoritmaları test edilmiştir. Çalışmalarının ana hedefi, SnortIDS, SuricataIDS, FL-SnortIDS ve FL-SuricataIDS konusundaki önceki araştırmalarıyla karşılaştırmalı olarak performans sergilemektir. En iyi sınıflandırma yaklaşımını bulmak için çapraz doğrulama (10 kat) ve Weka platformunda yapılan deney kullanılarak değerlendirmeler yapılmıştır. Deneysel çalışmalarda J48 özellikle SnortIDS, SuricataIDS ve FL-SnortIDS'te göre en iyi tespit oranını göstermiştir.

Bulut bilişim, kullanıcıların belirli bir işlem problemini çözmek için çok sayıda VM kullanmalarını sağlayarak isteğe bağlı işlem altyapısına erişim sağlayan bir modeldir. Bulutta çalışan yeni uygulamalar oluşturmada önce, olası uygulamaların farklı performanslarını/maliyetlerini değerlendirmek genellikle yararlıdır. Derfouf ve Eleuldj, CloudAnalyst kullanarak CIDS'in farklı koşullarını karşılaştırmışlardır. Simülasyon, kullanıcının saatlik ortalaması, toplam veri aktarımı, veri işleme ve yanıtlama süresi, VM maliyetleri vb. gibi birkaç kriter göz önünde bulundurularak yapılmıştır. Sonuçları önceki sistemleriyle karşılaştırmışlar (Derfouf & Eleuldj, 2017).

DDoS, bilgisayar ağındaki ünlü flooding saldırılarından biridir. Zekri ve ark, DDoS saldırılarını tespit etmek için karar ağacı (C4.5) olan makine öğrenme (ML) tabanlı bir algoritma geliştirmişler. Sistemlerinin ana hedefi imza saldırılarını tespit etmektir. Bu nedenle, sistem DDoS flooding tehditlerinin etkili ve otomatik olarak algılanması için karar ağacı oluşturur. Sonuçlarını diğer makine öğrenme algoritmalarıyla karşılaştırmışlar ve C4.5 in en iyi doğruluğu %98 oranıyla sağladığını göstermişlerdir (Zekri, El Kafhali, Aboutabit, & Saadi, 2017).

Javadpour ve ark. buluttaki algılama saldırılarının doğruluğunu artırabilecek yeni bir yöntem bulmak için makine öğrenme algoritmalarını test etmişlerdir. Yapay sinir ağı (NN), Rastgele Orman (RF), Karar Ağacı (DT) ve CART algoritmasını kullanarak saldırıları azaltmak için farklı makine öğrenme tabanlı özellik seçim algoritmaları uygulamışlardır. NN, %99.98 ile diğer algoritmalar arasında en yüksek doğruluğu elde etmiştir (Javadpour, Abharian, & Wang, 2017).

Hajimizaei ve Navimipour (Hajimirzaei & Navimipour, 2019) Yapay Arı Kolonisi, Çok Katmanlı Algılayıcı (MLP) ve Bulanık Kümeleme yönteminin kombinasyonuna dayanan yeni bir IDS yaklaşımı önermişlerdir. Verilerin sınıflandırıcısı olarak MLP ve MLP'yi eğitmek için de ABC kullanmışlardır. Diğer makine öğrenme yaklaşımlarıyla karşılaştırıldığında yüksek doğruluk sağlandığı görülmüştür.

Idhammad ve ark, bulut platformları için ML'ye dayanan bir Dağıtılmış İzinsiz Giriş Tespit Sistemi (DIDS) sunmuşlardır. Sistemleri, bulut sağlayıcısının uç ağ bileşenleri ile yan yana bulut içine yerleştirilmek üzere tasarlanmıştır. Bu, gelen ağ trafiğinin fiziksel katmanın kenar yönlendiricilerine yakalanmasına izin vermektedir. Her bulut yönlendiricisinde yakalanan ağ trafiğini ön işlemde geçirmek ve NB sınıflandırıcısını kullanarak bir anormallik algılama modülüne iletmek için kayan bir zaman penceresi algoritması kullanılmıştır. Ağ tıkanıklığı arttığında kullanılacak her anormallik algılama modülünün Hadoop tabanlı ve MapReduce tabanlı çekirdek sunucu düğümleri kümesi mevcuttur. Her zaman penceresi için yönlendiricinin her iki tarafındaki anormal ağ trafiği verileri merkezi bir depolama sunucusuyla senkronize edilir. Daha sonra, her bir saldırının türünü tespit etmek için çok sınıflı bir son sınıflandırma adımı gerçekleştirmek için rastgele orman temelli bir topluluk öğrenme sınıflandırıcısı uygulanır. Sistemi analiz etmek için google bulut ortamında ve veri seti olarak kullanılan CIDDs-001'de uygulamalar yapılmıştır. Sonuçlar RF sınıflandırıcı ile karşılaştırılmıştır. Sonuçlara göre %97 doğruluk ve %0.21 yanlış pozitif elde etmişlerdir (Idhammad, Afdel, & Belouch, 2018).

Bhamare ve ark. (2016), çalışmalarında, danışmanlı makine öğrenme modellerini (Karar ağacından J48, Bayesian networks'dan Naïve Bayes, regresyon tekniklerinden logistic regression (LR) ve SVM türleri: SVM-RBF, SVM-Polynomial, SVM-Linear) eğitmek ve test için UNSW veri kümesini kullanmışlardır. LR, SVM-RBF ve J48 iyi sonuçlar verdiği görülmüş sonrasında, bu modelleri ISOT veri kümesiyle test etmişlerdir. Bulgularını sunarak ve makine öğrenimi alanında bulut güvenliğine uygulanabilirliği için daha fazla araştırmanın gerekli olduğunu belirtmişlerdir. Dolayısıyla, danışmanlı ML tekniklerinin, bulut güvenliği bağlamında daha iyi performans göstermesi için önemli bir yeniden çalışma ihtiyacı duydukları sonucuna varmışlar ve gelecekte anomali tespiti, kümeleme ve/veya danışmansız makine öğrenimi gibi alanlarda daha fazla araştırmaya ihtiyaç duyulacağını belirtmişlerdir (Bhamare, Salman, Samaka, Erbad, & Jain, 2016).

Marwan ve ark. bulut ortamında veri işlemeyi güvence altına almak için ML tekniklerini kullanarak yeni bir yaklaşım önermişlerdir. Deneyde, görüntü piksellerini daha verimli bir şekilde sınıflandırmak için destek vektör makineleri (SVM) ve Bulanık C-means Kümeleme (FCM) algoritmaları uygulamışlardır (Marwan, Kartit, & Ouahmane, 2018).

Avdagic ve Hajdarevic (Avdagic & Hajdarevic, 2017) bulut saldırı tespit ve önleme sistemi için bir ana bilgisayar ve ağ çerçevesi sağlamak amacıyla Microsoft'un yeni teknolojilerini kullanmışlardır. Bu çalışmanın amacı, ağdaki anormallikleri tespit etmek ve bulut sistemleri tarafından üretilen büyük miktarda veri ve trafiği korumak için mimarinin kullanılmasını önermektir.

Swathi ve ark, Ağ Saldırı Tespiti Sistemleri (NIDS), Bulut Sistemler için Saldırı Tespiti Sistemleri ve Bulut sistemleri için ID için mevcut çeşitli teknikler hakkında ayrıntılı bir çalışma yapılmıştır. Son olarak, CIDD ve KDD CUP'99 veri setleri gibi araştırmacıların tekniklerini uygulamak ve analiz etmek için Bulut güvenliği için çeşitli veri setleri hakkında kapsamlı bir genel bakış sunmuşlardır.

Kullanılabilirlik, Bütünlük, Gizlilik ve mahremiyet, herhangi bir gerçek dünyada, özellikle de bulut bilişim ve akıllı ortamda temel sorunlar olarak kabul edilmektedir. IoT'le ilgili sistemlerdeki güvenlik açık deliklerinin yarattığı akıllı uygulamaları etkileyen güvenlik sorunları ve riskler bulunmaktadır, bu nedenle bu sistemlerin güvenliğini uygulamak, ağlara başarılı bilgi dağıtımının birinci önceliği haline gelmiştir. Ülker, E ve Nur, I . (ÜLKER & NUR, 2020), IoT ağına izinsiz girişleri savunmak ve güvence altına almak için yeni bir hibrit yaklaşım sunmuşlar. Özellik seçimi olarak Binary Grey Wolf Optimizer'i, sınıflandırma olarak Naive Bayes'i ve veri seti olarak BoT-

IoT'yi kullanmışlar. 99.15 % doğruluk elde etmişler ve diğer optimizasyon algoritmalarına kıyasla üstünlük göstermiştir.

He ve ark. makine öğrenme tekniklerine dayalı bulutta bir kaynak tarafı DoS saldırı tespit sistemi önermişler. Bu sistem, ağ paketlerinin harici ağa gönderilmesini önlemek için bulut sunucusu hipervizöründen ve sanal makinelerden gelen istatistiksel bilgileri kullanır. Dokuz makine öğrenme algoritmasını değerlendirmişler ve performanslarını dikkatlice karşılaştırmışlardır. Deneysel sonuçlar, dört DoS saldırısı tipinin %99.7'sinden fazlasının başarılı bir şekilde tespit edildiğini göstermiştir. Yaklaşımlarının performansı düşürmediği ve daha büyük DoS saldırılarına kolayca genişletilebileceğini ifade etmişlerdir (He, Zhang, & Lee, 2017).

Alzubi ve ark.(Alzubi, Anbar, Alqattan, Al-Betar, & Abdullah, 2019) tarafından değiştirilmiş ikili gri kurt optimizasyonu (MBGWO) adı verilen değiştirilmiş bir özellik seçim algoritması (FS) önerilmişlerdir. Önerilen algoritma, IDS'nin performansını artırmak için ikili gri kurt optimizasyonuna dayanmaktadır. Yeni özellik seçim algoritması optimum sayıda özellik seçmiştir. Önerilen algoritmayı değerlendirmek için, NSL-KDD ağ saldırısının veri setini kullanılmışlardır. Ek olarak, veri setini etkin bir şekilde sınıflandırmak için destek vektör makinesi kullanılmışlardır. Önerilen özellik seçim ve sınıflandırma algoritmaları, saldırıları tespit etmede IDS'nin performansını artırmıştır. Simülasyon sonuçları, önerilen algoritmanın izinsiz giriş tespitinin doğruluğunu artırdığını ve özelliklerin sayısını azalttığını göstermişlerdir.

Literatür incelendiğinde Bulut Bilişim Alanında Saldırı tespitinin önemli ve güncel bir alan olduğu görülmektedir. Farklı amaçlar ve tespitler için farklı algoritmalar kullanılsa da istenilen başarılarla ulaşmak için daha fazla çalışmaya ihtiyaç duyulduğu görülmüştür. Bu tezde bu amaçlara hizmet edebilecek bir makine öğrenmesi temelli yaklaşımın varlığı veya kullanılabilirliği gösterilmeye çalışılmıştır.

3. MATERYAL ve METOT

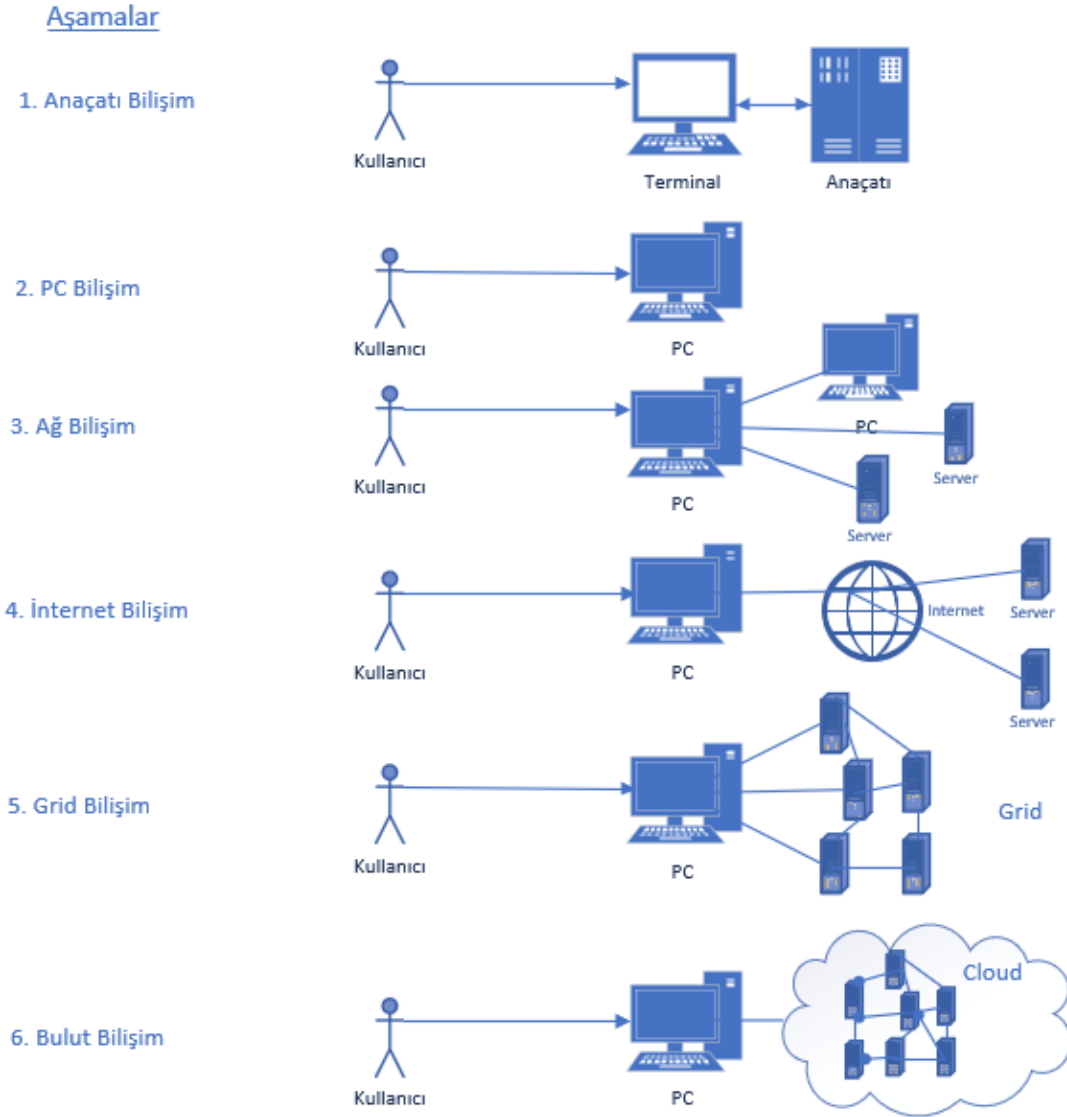
3.1. Bulut Bilişim

Bulut bilişim (Cloud computing - CC), veri ve uygulamaları kullanmak, depolamak ve işlemek için internet bağlantısına ve merkezi uzaktaki sunuculara bağlı olarak çalışan hizmet tabanlı bir teknolojidir ve kullanıcıların verilere daha ucuz bir fiyata ve daha hızlı bir şekilde erişmelerini sağlamaktadır. Yeni bir bilgi teknolojisi kuşağı olan bulut bilişimin literatürde çok fazla tanımı vardır, ancak genellikle ilgili kaynaklar tarafından ifade edilen ve en çok kabul edilen Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından tanınan bulut istihbaratları Bilgisayar ağları, veri depolama, sunucular, hizmetler ve uygulamalar vb. gibi yapılandırılabilir bilgi işlem kaynaklarının ortak koleksiyonlarına her zaman, uygun bir temelde ve her durumda, her zaman ve her yerde erişime izin veren bir modeldir (Mell & Grance, 2011). Bu kaynaklar, minimum idari çaba ve servis-servis sağlayıcı etkileşimi gerektiren kolayca temin edilebilir ve ağdan çıkarılabilir (Mell & Grance, 2011).

Bulut bilişim, Cliff'in yayınında (Cliff, 2010) aşağıdaki şekilde açıklanmıştır: Bulut bilişim, internet hizmetleri olarak sunulan uygulamaları ve bu hizmetleri sağlayan veri merkezlerinin donanım ve yazılım sistemlerini ifade etmektedir. Uzun süredir hizmetlerin kendisi de bir hizmet olarak yazılım olarak adlandırılmakta olduğu için bu terimin kullanımı tercih edilmektedir. Veri merkezi donanım ve yazılımı bir bulut olarak tanımlanmaktadır.

Bulut bilişim, işlemci ve depolama alanı gibi "kullandıkça öde" yaklaşımına dayanan bilişim kaynaklarını kiralama ilkesine dayanır ve kullanıma bağlı olarak ödeme planı çıkarılır. Altyapı ve uygulamalar birbirinden bağımsızdır ve tek bir merkezi kontrol ile herhangi bir yerden verilere erişilebilir, gerektiğinde hızlı bir şekilde kapasite ölçeklemesi yapılabilir, kaynak kullanımı kolayca kontrol edilebilir, raporlar ve kütük kayıtları alınabilir. Bulut sağlayıcıların sunduğu hizmetler, bireysel kullanıcılardan, küçük ve orta ölçekli işletmelerden kamuya ve özel şirketlere veya kuruluşlara kadar birçok farklı kullanıcıya yarar sağlayabilir. Bulut hizmetleri internet tabanlıdır. Böylece internete bağlanabilen herhangi bir cihaza sunulan servislere erişilebilir. Bulut bilişim organizasyonları için veri merkezi, yazılım, donanım ve enerji maliyetleri düşürülür ve sunulur. Kullanıcılar için her yerde veritabanlarına, bilgisayarlara, dizüstü bilgisayarlara, tabletlere ve akıllı telefonlara hızlı ve kesintisiz erişim anlamına gelir.

Bilgi teknolojilerinin tarihine bakıldığında, ilk başta anabilgisayarların büyük boyutlu bilgisayarlar olarak kullanıldığı görülmektedir. Eski teknolojiyi ve ana bilgisayar hesaplamayı göz önüne alarak, bulut bilgisayar birçok yeniliğe yol açmıştır. Yedekleme özelliğine sahip bu büyük tek sistemler, yüksek işlem hızında yüksek performans, güvenlik ve erişilebilirlik sağlayabildi. Ayrıca, ana bilgisayar sistemleri de sanallaştırma teknolojisinin temelidir. 1960'larda kullanılan anabilgisayar bilgisayarlarından bulut bilişime, geliştirme süreci (Furht & Escalante, 2010)'ten adapte edilmiş Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Bulut bilişim geliştirme aşamaları.

3.1.1. Bulut bilişimin sağladığı temel özellikler ve imkanlar

Bulut bilişimin birçok hizmet platformu ve altyapısı vardır. Bulut bilişimde, dinamik olarak sanallaştırılmış bilgisayarlar, esnek uygulamalar ve birbirleriyle ilişkili hizmetler, hesaplama, depolama ve ağ kaynakları vardır. NIST tarafından sağlanan bulut bilişim tanımından, günümüzde giderek daha popüler hale gelen bu yaklaşımın sağlayıcılara ve kullanıcılara beş temel özellik sağlaması gerekmektedir (Mell & Grance, 2011).

3.1.1.1. Talep üzerine hizmet

Bir kullanıcı, gerektiğinde sunucu işlem süresi ve ağ depolama alanı gibi bilgisayar kapasitesi miktarını otomatik olarak artırabilir veya azaltabilir. Bu tür kullanıcıların uygulamaları için gereken altyapının ne kadar iyi olacağını ve ne kadar kaynak kullanacaklarını kesin olarak analiz etmeye gerek yoktur.

3.1.1.2. Geniş ağ erişilebilirliği

Bulut bilişim kaynakları, dizüstü bilgisayarlar, tabletler ve akıllı telefonlar gibi farklı ağlardaki kullanıcılar tarafından kullanılabilir.

3.1.1.3. Kaynak havuzu

Sağlayıcının depolama alanı, işlem gücü, bellek ve ağ bant genişliği gibi bilgi kaynakları, dinamiğe bağlı olarak farklı fiziksel ve sanal kaynaklar üzerinde yeniden tahsis edilebilir ve kullanıcı istekleri, aynı zamanda çok sayıda kullanıcının aynı anda çoklu kullanım hizmetler verebilecek bir havuzda olması gerekmektedir. Kullanıcıların, mevcut kaynakların konumları üzerinde kontrol sahibi olmaları ve tam olarak nerede olduklarını bilmelerine gerekmemektedir. Hangi ülke ya da veri merkezlerinin yüksek düzeyde bilgi sahibi oldukları hakkında bir fikre sahip olmak, Hizmet Seviyesi Anlaşmasına (Service Level Agreement - SLA) bağlı olarak sağlayıcı tarafından önerilme fırsatına sahip olabilmektedir.

3.1.1.4. Anlık elastikiyet

Kullanıcı talebine veya sistem durumuna göre orantılı olarak ihtiyaç duyulan kaynaklar için bilişim kapasitesindeki kapasiteyi elastik/esnek bir şekilde artırabilir veya azaltabilir.

3.1.1.5. Ölçülebilir servis

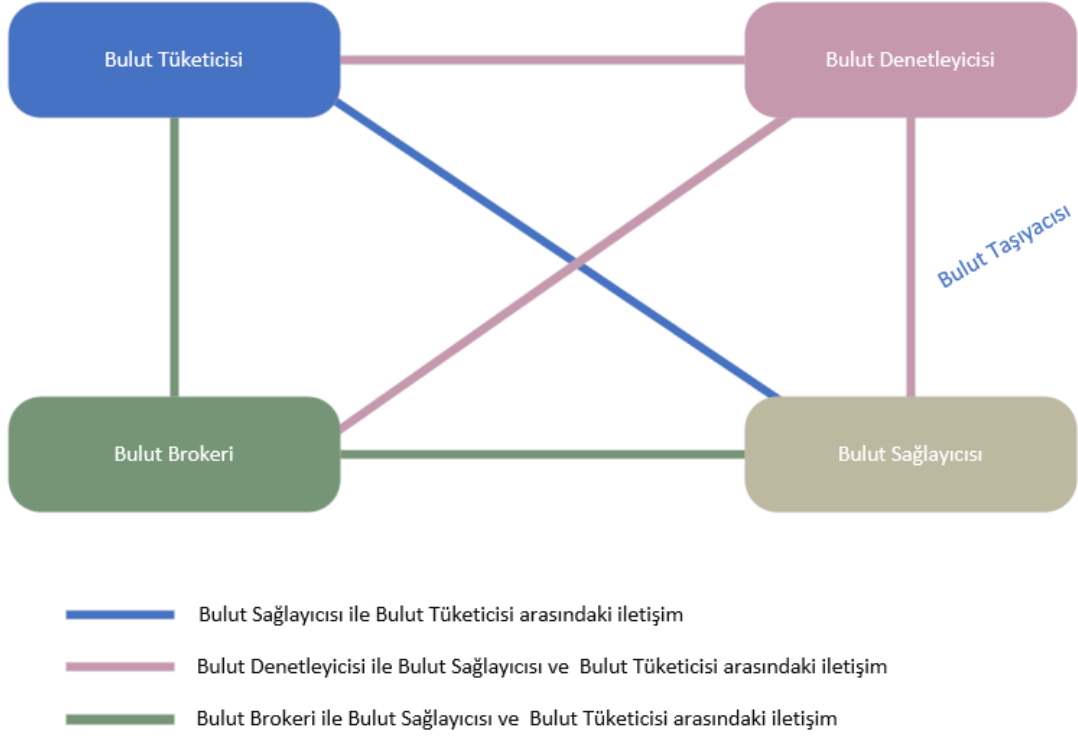
Bulut bilişimde kaynak kullanımı miktarı, hizmetin bir soyutlama katmanındaki (depolama alanı, işlemci gücü, bant genişliği ve aktif kullanıcı hesap sayısı gibi) bir ölçüm mekanizmasının yardımı ile otomatik olarak kontrol edilerek optimize edilebilir. Kaynak kullanım miktarları hem kullanıcılar hem de bulut sağlayıcılar için şeffaf bir şekilde kontrol edilebilir, izlenebilir ve raporlanabilir.

3.1.2. Bulut bilişim mimarisi

NIST, bulut bilişim mimarisinde beş temel aktör tanımlamaktadır. Bunlar tüketici, servis sağlayıcı, bulut denetleyicisi, bulut aracısı ve bulut taşıyıcısıdır. Her aktör, Şekil 3.2'de gösterildiği gibi bulut bilişimde bir görev/işlem gerçekleştiren bir varlıktır (Hogan, Liu, Sokol, & Tong, 2011).

3.1.2.1. Bulut tüketicisi (Cloud consumer)

Bulut tüketicisi, bulut bilişim tarafından desteklenen nihai paydaştır. Bulut tüketicisi, bulut sağlayıcısı tarafından sağlanan bulut hizmetlerini veya bulut sağlayıcısıyla iş ilişkisini kullanan bir kişiyi veya kuruluşu temsil etmektedir. Bulut sağlayıcısı tarafından sağlanan hizmetler arasında ihtiyaçlarına göre seçtiği uygun hizmeti talep eder, sağlayıcı ile bir hizmet sözleşmesi yapar ve hizmeti kullanır. Bu servisler servis sağlayıcıya sözleşmeye uygun olarak tahsil edilmektedir (Hogan et al., 2011).



Şekil 3.2. Bulut bilişimde aktörler arasındaki etkileşimler

3.1.2.2. Bulut sağlayıcısı (Cloud provider)

Servis sağlayıcı, bireyler, kuruluşlar veya tüketiciler tarafından kullanılan servislerin geliştirilmesinden sorumlu bir varlık olabilmektedir. Gerekli yazılım/platform/altyapı hizmetlerini yöneterek, hizmetlerin sürekliliği için gerekli teknik altyapıyı oluşturur ve belirtilen hizmet seviyelerini, özellikle de güvenlik ve gizliliği sağlamaktadır.

3.1.2.3. Bulut aracısı (Cloud broker)

Bulut bilişimin yönetimi, tüketicileri açısından oldukça karmaşıktır. Tüketici, doğrudan servis sağlayıcı ile iletişim kurarak değil, bir servis temsilcisinin yardımıyla bulut servislerini kullanabilmektedir. Servis aracısı, tüketici ile servis sağlayıcı arasındaki ilişkiyi düzenleyen bulut servislerinin performansını ve kullanılabilirliğini yöneten bir varlıktır.

3.1.2.4. Bulut denetleyicisi (Cloud auditor)

Bulut denetleyicisi, bilgi sistemi işlemleriyle bulut hizmetlerinin performansına ve güvenlik uygulamalarına bağımsız olarak bağlanabilen ve hizmet sağlayıcısını güvenlik denetimleri, gizlilik uygulamaları ve performans gibi ölçütler dahilinde denetleyen üçüncü taraf bir kişi veya kuruluştur.

3.1.2.5. Bulut taşıyıcısı (Cloud courier)

Bulut taşıyıcısı, bulut sağlayıcıları ve bulut tüketicileri arasındaki bağlantıları ve transferleri gerçekleştirip, tüketicilere ağ, iletişim altyapısı ve diğer cihazlar üzerinden servislere erişim sağlamaktadır. Tüketiciler, bulut hizmetlerine masaüstü veya dizüstü bilgisayarlar ve mobil cihazlar gibi ağ erişim cihazlarıyla erişebilmektedir. Bulut hizmetlerinin dağıtımı normalde, ağ ve iletişim altyapısının veya yüksek kapasiteli depolama tesislerinin kullanıldığı fiziksel iletişim ortamlarına sahip iletişim araçları aracılığıyla gerçekleştirilebilmektedir. Bulut sağlayıcısının, bulut taşıyıcısının hizmetlerini tutarlı bir şekilde sunması için SLA'lar hazırlaması gerekir ve bulut taşıyıcılarının, bulut sağlayıcısına adanmış bir altyapıya sahip, bulut sağlayıcısı ve bulut tüketicisi arasında şifreli bağlantılar kurabilmesi gerekmektedir.

3.1.3. Bulut bilişim servis modelleri

Bulut bilişimde farklı hizmet modelleri, son kullanıcılar, uygulama geliştiricileri veya sistem yöneticileri gibi farklı kullanıcılar için farklı hizmetler sunmaktadır. Bu hizmetler üç temel hizmet sunum modelinden oluşur: hizmet olarak yazılım (Software as service – SaaS), hizmet olarak platform (Platform as service – PaaS) ve hizmet olarak altyapı (Infrastructure as service – IaaS) (Mell & Grance, 2011).

3.1.3.1. Hizmet olarak yazılım (SaaS)

SaaS, sunucudaki yazılım uygulamaları bulut sağlayıcısı tarafından birden fazla kişi veya kuruluş tarafından kullanılabilir. Bu servis modeliyle, kullanıcılar aynı kullanım ve daha düşük lisans ve yazılım maliyetlerine sahiptirler, bu nedenle bulut bilişim hizmetlerini daha düşük maliyetle kullanılmaktadırlar. Kullanıcı, yazılımın yüklenmesi, lisanslanması ve bakımı gibi sorunlarla ilgilenmez, bu nedenle bu görevler

için maliyet ve zaman kaybı otomatik olarak kaldırılır. Kullanılan uygulamalara, internete bağlı herhangi bir bilgisayar veya mobil erişim sağlanabilir ve kullanılabilir. Bu, uygulamaların güncel, bakımı yapılabilir ve kesintisiz yüksek kullanılabilirliğe sahip olmasını sağlamaktadır. Kurumdaki veri merkezi yerine bu hizmeti sağlayan bulut sağlayıcıların veri merkezinde yer alan uygulamalarla daha güvenli ve hızlı bir yapı oluşturulabilmektedir.

3.1.3.2. Hizmet olarak platform (PaaS)

Bu model, kullanıcıya uygulama geliştirme, uygulama platformu ve gerekli tamamlayıcı hizmetler ve teknolojik altyapı sağlayan bir bulut sağlayıcısıdır. Oluşturduğu uygulamalar dışında, kullanıcının platform altyapısını oluşturan bileşenler üzerinde herhangi bir yönetimi ve kontrolü yoktur. Kullanıcılar, bu hizmeti sunan bulut servis sağlayıcılarından uygulamalar geliştirmek için gereken işletim sistemini, platformu ve araçları alabilmektedir. Bu noktada, kullanıcılar gerekli işletim sistemini, uygulamaları ve donanımı sağlamak için güçlük çekmeden doğrudan uygulamalar geliştirmeye başlayabilmektedir.

3.1.3.3. Hizmet olarak altyapı (IaaS)

Bu özellik, kullanıcıların bilişim kaynaklarına, depolama alanlarına, ağa ve sağlayıcının yetenekleri kapsamındaki diğer temel bilişim kaynaklarına erişmelerini ve işletim sistemleri ve uygulamaları da dâhil olmak üzere her türlü yazılımı kullanmalarını sağlamaktadır. Kullanıcıların bulut altyapısını kontrol etme ve yönetme zorunluluğu yoktur, yalnızca işletim sistemi, depolama ve dağıtılmış yazılımlar ve kullanılacak ağ bileşenlerinin seçimi gibi sınırlı yapılandırma alanları üzerinde söz sahibidir.

IaaS'de, fiziksel altyapı, depolama, ağ iletişimi ve temel bilgi işlem kaynakları, barındırma ortamları ve IaaS tüketicileri için gereken bulut altyapısını oluşturmak için sağlayıcı tarafından sağlanmaktadır. Tüketiciler, uygulamaları dağıtırken ve bu uygulamaları çalıştırırken, hosting ortamında diğer hizmetlerden daha fazla yetkiye sahip olmak için fiziksel sunucu, ağ, depolama ve hipervizör gibi bulut altyapısındaki varlıklar üzerinde herhangi bir işlem gerçekleştirememektedir. IaaS sağlayıcıları fiziksel ve sanal mimarlar aracılığıyla hizmet vermektedir. Sanallaştırma teknolojisi ile fiziksel sunucularda kendi yapılarında sanal bir mimari işletmektedirler. Bu şekilde, izole edilmiş

sanal sunucular oluşturulabilir, böylece IaaS sağlayıcıları kullanıcılar için sahip oldukları fiziksel sunuculardan daha fazla sayıda sanal sunucuya hizmet verebilmektedir. Tüketicilere, gereksinimlerine uygun olarak gerekli yazılımları kullanabilecekleri ve dağıtabilecekleri VM'lere, ağ depolama birimlerine, ağ altyapısı bileşenlerine ve diğer temel bilgi kaynaklarına erişim sağlanmaktadır. IaaS tüketicileri genellikle sistem geliştiricileri, sistemleri ve IT yöneticileridir. IaaS modelinin en büyük avantajlarından biri, büyük ölçekli kurum ve kuruluşların donanımaya yatırım yapmasının gerekli olmamasıdır. Kurumlar için kazanılması en büyük kazanım, ölçeğe göre ödeme ve esneklik gerekliliği ve sürekli mevcut altyapı için yeni teknolojilere yatırım yapma zorunluluğunun da dışlanmış olmasıdır. IaaS modeliyle yönetimin basitliği ve dinamik doğası da bu başarıları büyük ölçüde artırmaktadır (Rimal, Jukan, Katsaros, & Goeleven, 2011).

3.1.4. Bulut bilişim dağıtım modelleri

Hizmet Sağlayıcı, bulut bilişim sistemini yönetim, maliyet ve güvenlik gibi nedenlerle yönetebilir veya üçüncü taraflara veya kuruluşlara atayabilir. İhtiyaçlara ve gereksinimlere bağlı olarak şekillendirilebilen ve esnek biçimde bağlanabilen dört tür bulut bilişim dağıtım modeli vardır.

3.1.4.1. Genel bulut (Public cloud)

Herhangi bir kişi veya kuruluş tarafından işletilen ve üçüncü taraf kullanıcılara bulut hizmetleri sağlayan bir bulut bilişim çözümüdür. Bulut altyapısı herkese açık ve genel olarak internet iletişim altyapısı olarak kullanılmaktadır. Faaliyetler; görev dağılımı, işletme, organize edilme, yönetilme ve işletilme şeklinde gerçekleştirilebilmektedir.

3.1.4.2. Özel bulut (Private cloud)

Organizasyonlara özel veri merkezlerine inşa edilmiş bir bulut yapısıdır. Bu modelde kurumlar bulut altyapısını kendi binalarında barındırmakta ve kendi hizmetlerini sunmaktadır. Özel bulut sayesinde, kuruluşlar kendi bilgi altyapılarını esnek ve düşük bir maliyetle işletebilir ve kendi güvenlik politikalarını yönetebilmektedir. Veri gizliliği ve

güvenliğinin en üst seviyede tutulması gereken durumlar için tercih edilen geleneksel bir modeldir. Özel bulut, kimseye açıklık verilmeksizin, yalnızca güvenlik duvarı arkasındaki dâhili kullanıcılara hizmet etmektedir.

3.1.4.3. Topluluk bulutu (Community cloud)

Kuruluşun ve belirli bir topluluğu oluşturan kişilerin gerektirdiği görevler, güvenlik gereklilikleri bağlamında tesis edilir. Politikalar, bir veya daha fazla kuruluş veya kişi tarafından oluşturulabilir, yönetilebilir ve işletilebilir. Bunların tümü üçüncü şahıs bir şirket veya bağlı kuruluşları tarafından gerçekleştirilebilmektedir. Topluluk bulutları, güvenlik ve yasal gereklilikler için dernekler veya kuruluşlar tarafından sıklıkla kullanılmaktadır. Bu, birbirlerinin kaynaklarına erişmelerine ve aralarında veri alışverişine olanak sağlamaktadır (Rimal et al., 2011).

3.1.4.4. Hibrit bulut (Hybrid cloud)

Bulut bilişim altyapısının, diğer konumlandırma modellerinden en az ikisinin birlikte kullanarak oluşturulduğu bir yapıdır. Gerekli görevler, güvenlik gereksinimleri, politikaları ve teknolojik imkânlar dâhilinde yük dağılımı gibi veri ve uygulama taşınabilirliğini sağlamak için yapılandırılmıştır. Bu bulut yapılarının en önemli sorunu, karmaşıklığı gizlemek için farklı mimarilerde farklı hizmetlerin birlikte çalışabilirliğini sağlamaktır. Bu gibi durumlarda, üst düzey veri aktarımı ve senkronizasyon problemleriyle karşılaşılabilir (Rimal et al., 2011).

3.1.5. Bulut bilişim avantajları

Verimlilik: Kendi IT sistemlerini işleten kuruluşların, ihtiyaç duydukları donanım ve yazılım altyapılarını ve ayrıca zaman ve maliyet açısından verimsiz bir şekilde çalışan personeli ayarlamaları gerekmektedir. Bulut bilişim bu verimsizliği zaman ve maliyet açısından ortadan kaldırmaktadır. Temel çatıdan kaynaklanabilecek aşırı işlem gücü veya depolama kapasitesi, kaynakların verimli kullanılmamasına neden olabilmektedir. Birçok kurumsal veri merkezi sahip oldukları kaynak kapasitesinin yüzde 50'sinden daha azını kullanmaktadır.

Hazır bulunurluk: Genel olarak, bulut bilişimi savunan kişilerce, yüksek donanım gücüne sahip altyapı veri merkezlerinde çalışan sistemlerin kullanılabilirliğinin daha yüksek olduğu kabul görmektedir (Leavitt, 2009).

Uygulama uyumluluğu ve desteği: İyi tasarlanmış bulut bilişim sistemlerinde bulunan karmaşık yapılarla güçlü uygulamalara entegre olacak birçok hizmet daha istikrarlı bir şekilde çalışacaktır. *Esneklik:* bulut bilişim sağlayıcılarının çoğu, yürüttükleri hizmetler ile ilgili anlaşmalara ihtiyaç duymamaktadır. Bu nedenle, yeni hizmetleri ve ürünleri test etmek için ek servislere ihtiyaç duyulduğunda, bulut bilişim etkili bir çözümdür. Bulut bilişimin esneklik açısından faydaları aşağıdaki gibidir:

- Endişelenmeden sunucuları güncelleme
- Yazılım güncellemelerinin kolayca yapılabilmesi
- İstenilen ölçekte kaynakların artırılabilmesi
- Donanım bağımsızlığı
- Yeni hizmet ve teknolojilerin otomatik olarak sağlanması
- Bakım detayları yerine yeniliğe odaklanabilme

3.1.5.1. Düşük maliyetli donanım

Tarayıcılarda çalışan uygulamaları kullanmak için yüksek performanslı donanım gerekmemektedir (YÜKSEL, 2012). Bu tür uygulamaları çalıştırmak için akıllı telefonlar bile yeterlidir. Bunun nedeni, uygulamalarınızın bulut bilgi işlem sistemleri üzerinde çalışıyor olmasıdır. Minimum kaynak tüketimi ile günümüz teknolojisinde kullanılan en güçlü kişisel bilgisayarların performanslarına ulaşılabilir. Bu nedenle, düşük kapasiteli sabit disk, bellek veya işlemciye sahip bulut bilişim kullanıcıları performans kaybına neden olmamaktadır. Sonuç olarak, düşük maliyetli bilgi sistemlerinin kullanılması maliyetleri düşürmektedir.

3.1.5.2. Düşük yazılım maliyetli

Bulut bilişim sistemlerinde, her kullanıcı bilgisayar için ayrı bir lisans yazılımının kullanılmasını gerek kalmamakta ve uygulama servisleri aracılığıyla kullanıcıların ihtiyaç duyduğu yazılımlara erişim sağlanmaktadır. Kurumsal kullanıcıların boyutu büyüdükçe, yazılımın dağıtılması ve yönetimi ek bir yük haline gelmektedir. Ancak, bulut bilişim yetenekleriyle birlikte kullanılacak bir yazılım hizmetinin kullanılabilirliği

nedeniyle, geleneksel bilgisayar sistemlerinin maliyetleri düşmektedir. Satın alınacak yazılım bir hizmet olarak kiralanabilir ve maliyetler buna göre düşürülmektedir.

3.1.5.3. Performans

Bulut bilişimin önemli kullanımlarından biri, yoğun işlem gücü gerektiren uygulamalardır. Sanallaştırma sayesinde, bulut bilişim altyapısının aynı anda yüksek performanslı çok sayıda işlemi gerçekleştirebilmesi sağlanmaktadır.

3.1.5.4. Güncelleme

Uygulamaların yeni sürümleri dağıtıldığında veya bu yazılımların güncellemeleri yayınlandığında, en güncel sürüm bulut bilişimde aynı anda kullanılabilir. Kullanıcıların bunlar için ek ücret ödemeleri gerekmez ve bu işlemleri gerçekleştirmek için sistem yöneticisi veya teknisyene ihtiyaçları gerekmemektedir.

3.1.5.5. Yüksek depolama kapasitesi

Kişisel bilgisayarlarda kullanılan 1.5 terabayt sabit diskli sunucu bilgisayarlarda kullanılan ağ kapasitesine sahip 1 petabayt kapasiteli sabit sürücünün depolama kapasiteleri aynı boyutlarda değildir.

3.1.5.6. Veri güvenliği

Kişisel bilgisayarların donanımı kolayca başarısız olabilir ve en kötü durumda, sabit sürücüler kullanılamaz ve verilerin içeriğine erişilemez hale gelebilmektedir. Bulut bilişim sistemlerinde, kullanıcıların verileri bulut yedeklemesinde depolanmakta, olağanüstü durum kurtarma önlemleri alınmakta ve veri kaybı olmamaktadır.

3.1.5.7. İşletim sistemleri arasında uyum

Farklı çekirdekli mimarilere sahip işletim sistemleri arasında ağ bağlantısı ve veri değişimi nispeten kolay değildir. Bu, işletim sisteminden bağımsız olarak, Windows işletim sistemine sahip bir bilgisayar ile MacOS veya Linux dağıtımına sahip bir

bilgisayar arasında veri aktarımında olası sorunların başından beri, işletim sisteminden bağımsız olarak verilerin varlığına neden olabilmektedir.

3.1.5.8. Dosya formatının uygunluğu

Geleneksel IT sistemlerinde, bir belge bilgisayarın dışında açıldığında nasıl bir tabloyla karşı karşıya geleceği belirsizdir. Yazılım veya işletim sistemi belgesinin sürümü arasındaki fark açık değildir. Web tabanlı bir belge işlemcisi hazırlanan bir belge, erişilen her cihazda aynı şekilde görüntülenebilmektedir. Bununla birlikte, geleneksel bir IT sisteminde, bir belge işlemcisi yazılımında hazırlanan bir belge aynı yazılımın eski sürümlerinde görüntülenemeyebilmekte ve ek yazılım veya eklerin yüklenmesi gerekebilmektedir.

3.1.5.9. Grup çalışması

Bulut bilişim sayesinde, aynı belgede birden fazla kişi düzenleme yapabilir. Geleneksel IT sistemlerinin aksine, aynı anda aynı belgede birden fazla kişinin çalışmasına izin verilmektedir. Dosyalar üzerinde herhangi bir işlem yapmak isteyen bir kişi en güncel çevrimiçi belgeye erişebilmektedir.

3.1.6. Bulut bilişim dezavantajları

Bulut bilişimde bireysel veya kurumsal olabilecek kullanıcılar aşağıda verilen bazı zorluklarla karşılaşabilmektedir:

3.1.6.1. Mevcut uygulamanın bulut altyapısına taşınmasının zorlukları

Bulut bilişim sağlayıcılarına göre farklılık gösterse de, bulut bilişim altyapısına aktarılan uygulamaların önemli miktarda yeniden yazılması gerektiği belirtilmektedir (Kwasniewski & Puig, 2011). Geleneksel mimariye göre geliştirilen bir yazılım, dağıtılmış bir mimariye aktarılırken büyük bir şekilde değiştirilmelidir. Ancak, grid bilişim gibi dağıtılmış bir mimari için geliştirilen yazılımı bulut bilişim altyapısına taşımak nispeten kolay olabilmektedir. Bulut sağlayıcıları, güvenlik nedenleriyle programlama dillerindeki bazı işlevleri devre dışı bırakabilir. Bu devre dışı özellikleri

kullanan yazılımın değiştirilmesi gerekecektir. Ek olarak, programlama dilinde bu farklılıkları anlama ve hesaba katma sorumluluğu servis alıcıya bırakılmıştır (ERGİN, Gerede, & Korkmaz, 2012).

3.1.6.2. Deneyimsiz bulut operatörü / sağlayıcısı

Servis sağlayıcı bulut bilişim sisteminin altyapısı, platformu veya servisleri üzerinde gerekli bakımı yapmazsa, arıza durumunda veriler tekrar erişilemeyecek şekilde kaybedilecektir. Veriler hizmetten daha önemlidir.

3.1.6.3. Sistem güncellemeleri

Bulut bilgi işlem sistemlerinde kullanılan yazılım, bulut altyapısı güncellendiğinde sorunlarla karşılaşabilir. Servis sağlayıcısı, deneyimli çalışanların sağladığı sistem ve yazılım güncellemeleri / yükseltmeleri avantajını bir dezavantaja dönüştürmemelidir.

3.1.6.4. Güvenlik ve Gizlilik

Bulut bilişimi kullanan kurumlar ve kuruluşlar hala güvenlik konusunda endişelidir. Mevcut tüm bilgilerin mevcudiyeti erişilebilirlik açısından büyük bir avantaj olsa da, kötü niyetli servis sağlayıcıların veya sistem yöneticilerinin bu bilgileri kötü amaçlar için kullanması veya bu bilgilere erişildiğinde hiç kullanmaması olasıdır.

3.1.6.5. Bant genişliği maliyeti

Kullanılan cihazın ağ bağlantısı yoksa, web tabanlı bir uygulamaya veya hizmete erişmek mümkün değildir ve gerekli gereksinimlerin üstesinden gelinememektedir. Bulut bilişim sayesinde, kullanıcılar donanım ve yazılım maliyetlerinde tasarruf sağlarken, bazen daha yüksek ağ bant genişliği maliyetleriyle karşılaşmaktadır. Bant genişliği maliyeti, verileri erişilemeyen küçük internet tabanlı uygulamalar için düşük olabilir, ancak veri yoğun uygulamalar için önemli ölçüde artmaktadır. Tarayıcı tabanlı uygulamaların yüksek bant genişliği nedeniyle, bulut bilişim hizmetleri ve uygulamaları, çevirmeli bağlantılar veya Genel Paket Radyo Servisi (GPRS) gibi düşük bant genişliği

bağlantılarında iyi performans gösterememektedir. Ayrıca, geniş bant ağ bağlantısının asimetrik indirme ve yükleme oranları nedeniyle, yedekleme, güncelleme ve indirme işlemleri geleneksel IT sistemlerinden daha yavaş olabilmektedir. Geniş bantlı bir ağ bağlantınız olsa bile, kullanılan servis, sunucunun alınan hizmetin veya servisin desteklemeyecek kadar yoğun olması nedeniyle yavaş olabilmektedir.

3.2. Bilgi Güvenliğine Yönelik Tehditler ve Saldırı Tespiti

Bu bölümde önce, günlük sistemlerde meydana gelen bazı önemli tehditler/saldırıları gözden geçirilecektir. Daha sonra saldırı tespit teknikleri ele alınacaktır.

3.2.1. Bilgi güvenliğine yönelik tehditler

Bilginin taşıma ve depolama ortamındaki değişiklikler, bilgiyi tehdit eden unsurları değiştirmiştir. Bu tehditler kağıt yerine bilgisayar ürünlerinin kullanımından kaynaklanmaktadır. Ek olarak, bilgi hareketliliğindeki artış nedeniyle, en az medya depolarının güvenliği kadar medyanın nihai sonucunun güvenliği de önem kazanmıştır.

Bugünün teknolojisindeki bilgi miktarının her 18 ayda bir iki katına çıkması hem bu devasa miktarda bilginin korunmasını zorlaştırmakta hem de güvenliğin önemini artırmaktadır. Elektronik ortamda üretilen, depolanan ve aktarılan bilgilerin neredeyse hepsinin elektronik ortamda saklanması, “bilgi güvenliği” kavramının “bilişim güvenliği” olarak anlaşılmasını sağlamıştır. Bilgi güvenliğine yönelik tehditler aşağıdaki şekilde tanımlanmaktadır:

3.2.1.1. İç saldırılar

Sistem içerisinde, kurumda veya ağ içinde kalmaya yetkili olanların kendi yetkilerinden daha fazla hakka sahip olmaları durumunda, sistemin normal işleyişinin tahrip edilmesi veya üst düzey sistem araçlarına ulaşmalarının bir sonucu olarak saldırılar oluşmaktadır.

3.2.1.2. Virüs

Otomatik olarak kendini kopyalayan, bir taşıyıcı programa ihtiyaç duyan ve kendisini bir sistemden diğerine yayan bir kod parçasıdır. Ana amaç bilgisayar kaynaklarına zarar vermektir.

3.2.1.3. Programlama dillerinin zayıf yönlerini kullanarak saldırılar

Programlama dillerinin özelliklerinden yararlanan uzman programcılar, dizginin formatını değiştirerek, verileri işlevlere göndermek için özel karakterler kullanarak belleğe yetkisiz erişime neden olurlar (Canli, 2009).

3.2.1.4. Reklam yazılımı

Doğrudan bilgisayara zarar vermeyen ve kullanıcıyı istemeden ağa maruz bırakan yazılımlardır. Çok sayıda işlem olması durumunda, kullanıcının sistem performansını önemli ölçüde azaltır.

3.2.1.5. Casus yazılımı

Kullanıcıların verilerini habersiz bir şekilde yazılım sahiplerine gönderen, bilgisayar üzerinden kurulan bir yazılımdır. Virüslerin aksine kendilerini kopyalamazlar, sistemdeki görevi tek bir kod bloğunda yaparlar.

3.2.1.6. Solucan

Kendilerini çok sayıda yeniden üretebilen, bir programdan bağımsız olarak çalışan ve ağa yayılabilen programlardır.

3.2.1.7. Truva atları

Truva atları, bilgisayar kullanıcılarına başka faydalı işlevler sunan, saldırgan tarafından yazılan özel kodlara sahip olan, bilgisayardaki kaynaklara ve bu sitedeki verilere ulaşan programlardır.

3.2.1.8. Çeviriciler

Bunlar, çevrilen şebekelerde özel kod numaralarını arayan ve kullanıcının daha yüksek bir fatura almasına neden olan, çevrilen numaralarda para kazancı sağlayan kötü amaçlı yazılımlardır.

3.2.1.9. Kök kullanıcı takımlar

Bu yazılımın amacı, sistemdeki etkin görevler, işlemler ve sistemle ilgili bilgilerle ilgili işletim sisteminden gizlenmektir.

3.2.1.10. Araç çubukları

İnternet tarayıcılarına yüklenen yazılımlardır. Bu kendi kendini kopyalayan yazılımlar genellikle kullanıcı bilinci olmadan yüklenir ve bir tür reklam yazılımı gibi davranır.

3.2.1.11. Hijackers

Gerçek bir sitenin web sitesine benzer bir web sitesi kurarak ve gerçek web sitesini içerik olarak kopyalayarak sık sık banka veya kullanıcı hesap bilgilerini almaya çalışan kullanıcıyı aldatan programlardır.

3.2.1.12. Keyloggers

Bilgisayara gizlice yüklenen yazılımdır, tüm tuş vuruşlarını, yazılımın ayarlandığı noktaya gönderir. Kullanıcı hakkındaki tüm şifre bilgileri ve aktif yazılımla yazışmalar çalınabilir.

3.2.1.13. IP gizleme

IP gizleme, bilgisayar korsanlarının kendilerini farklı bir kullanıcı olarak tanımlamaya veya geçersiz e-posta adresleriyle farklı şekillerde tanımlamaya ve aldatıcı etkinliklerin sonucu olarak gerçek kimliklerini gizlemeye çalıştıkları bir güvenlik tehdididir.

3.2.1.14. SPAM - Mesaj sađanađı

E-postalar kullanıcıya kendi istekleri dışında gönderilir. Amaç, genellikle reklam olan e-postaların, kullanıcıların e-posta kutularını doldurması ve zaman kaybına neden olmasıdır. Ayrıca e-posta veritabanlarına ağır yük getirmektedirler.

3.2.1.15. Arka kapı

Bu, saldırganların bir düğüme yerleştikten sonra kullanıcının haberi olmayan DDoS saldırıları için zombi olarak kullandığı pasif saldırı türüdür. Truva atları ve diğer benzer yazılımlar düğümü devralmaya yardımcı olur. Sistem bir zombi olarak yakalandığında, üzerindeki tüm verilere kolayca erişilebilir (Modi et al., 2013).

3.2.1.16. Ortadaki adam

Saldırganın iki uç iletişimin arasında konumlandırıldığı, tüm iletişim köprüsü üzerine kurulu olduğu, böylece tüm iletişim verilerinin elde edildiği saldırılardır. Çoğu, MAC (Medium Access Control) adreslerini değiştirir.

3.2.1.17. Hizmet reddi saldırıları (Denial of Service DoS)

Bu saldırılar, servis sağlayıcı kasıtlı olarak sunucuya aşırı bir talepte bulunurken, sunucuyu gerçek talebe cevap veremez hale getirir.

3.2.1.18. Dinleme veya yakalama (Sniffing)

Dinleme (Sniffing), paket dinleyicileri kullanarak bir bilgisayar ağından geçen tüm veri paketlerini izleme ve yakalama işlemidir. Paket Dinleyiciler, ağ yöneticileri tarafından ağlarından geçen veri trafiğini takip etmek için kullanılmakta ve bunlara ağ protokolü çözümleyicileri denir. Aynı şekilde, kötü niyetli saldırganlar, bir ağdaki veri paketlerini yakalamak için bu paket dinleme araçlarını kullanmaktadır.

Sniffing saldırılar, kullanıcı adı ve şifreleri almayı, bankayla ilgili / işlemle ilgili bilgileri çalmayı, e-posta ve sohbet mesajlarını gözetlemeyi ve kimlik hırsızlığını gibi motive etmektedir.

3.2.2. Saldırı tespiti

IT dünyasında güvenlik tehditlerinin sayısındaki ve çeşitliliğindeki hızlı artış ile güvenlik mekanizmalarında hızlı gelişmeler olmuştur. Kimlik doğrulama ve erişim kontrolü gibi güvenlik mekanizmaları ilk olarak bilgisayarları korumak ve yetkisiz kullanıcıların yetkisiz sistemlere erişimini veya bu sistemlere müdahale etmesini önlemek için geliştirilmiştir. İnternetin yaygın kullanımı ile birlikte IT sistemlerine yönelik güvenlik tehditlerinde ve saldırıların yayılmasında önemli artışlar olmuştur. Tehditler ve saldırılar nedeniyle yeni mekanizmalar geliştirilmelidir. Bilgi sistemlerinde güvenliği sağlamak için güvenlik duvarları, güvenlik açığı tarayıcıları ve saldırı tespit sistemleri de kullanılır.

Bu güvenlik mekanizmalarından hiçbirinin, tek başına kullanımının güvenlik için tamamen yeterli olmadığı düşünülmektedir; Çünkü her biri farklı açılardan güvenlik konularına odaklanmıştır. Sistemde güvenliğin sağlanması, bu mekanizmaların birbirlerini desteklemek için birlikte kullanılmasını gerektirir. Bulut bilişim sistemleri ek olarak güvenliği sağlamak için çeşitli yönetim modellerine odaklanmıştır (Kumar et al., 2012). Bu modeller genellikle Saldırı Tespit Sistemi, Güvenli Hesaplama, Veri Şifreleme gibi mekanizmaları içerir.

Bir kaynağın veya verinin güvenilirliğini, bütünlüğünü, gizliliğini veya erişilebilirliğini önleyen tüm eylemler bir saldırı olarak tanımlanır. Saldırı tespiti, bilgi sistemi veya iletişim ağındaki alt sistemleri olumsuz yönde etkileyebilecek zararlı davranışlara neden olabilecek şüpheli olayları tespit eden, her türlü paketi ve verileri analiz ederek gerçekleştirilen adımların bir birleşimidir. Gerçek zamanlı veya sonrasında meydana gelen saldırıları tespit etmek için alınan tüm önlemlere rağmen, bilişim sistemlerinde hala yeni tespit mekanizmaları gerekmektedir (Scarfone & Mell, 2012; Tyler, 2009).

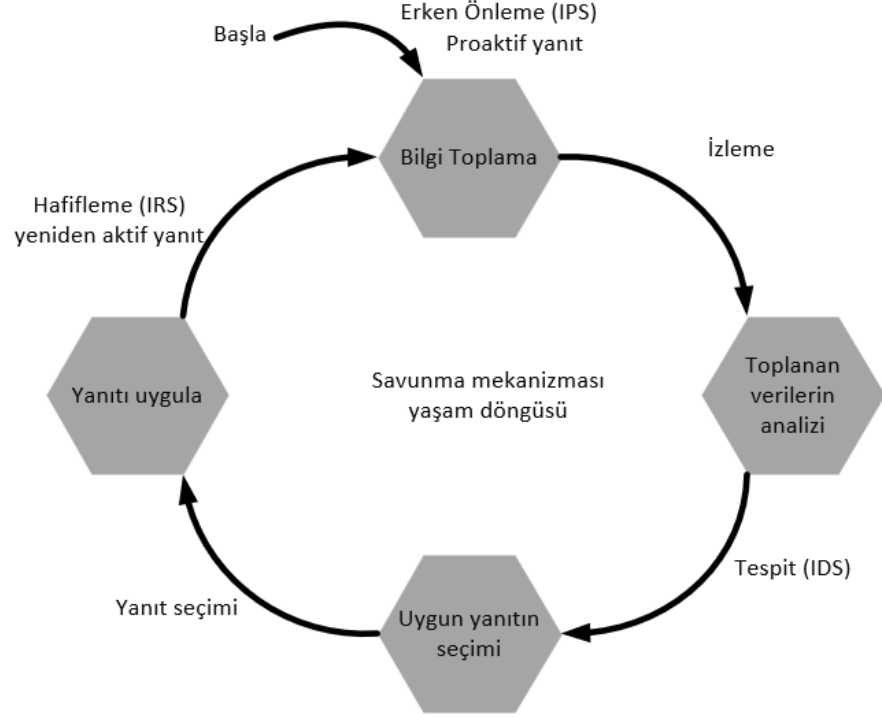
Saldırı tespit sistemleri, saldırı tespit işlemlerini gerçekleştiren ve tespit edilen bilgileri yedekleyen, bu yönde alarmlar üreten veya önceden tanımlanmış bir dizi işlem gerçekleştiren sistemlerdir. Gözlemlenen bileşen, IT kaynakları dâhil olmak üzere her türlü donanım veya yazılım olabilir. Yakalanan her şüpheli olay, saldırı girişimi anlamına

gelmez. Beklenmeyen bazı olaylar nadiren gerçekleşir ve saldırı olup olmadıklarına karar vermek önemlidir. Başka bir deyişle, Bir Saldırı Tespit Sistemi (Intrusion Detection System, IDS), kötü amaçlı faaliyetlerin ve bilgisayar güvenlik politikalarının, kabul edilebilir kullanım politikalarının veya raporlama için yazılımı veya donanım sistemini otomatikleştiren bu sistemlerde meydana gelen standart güvenlik politikası ihlallerinin bir analizidir (Scarfone & Mell, 2012).

Bir IDS, korunan sistemden veri toplayan ve sistem yöneticilerini, izinsiz veri kaydedilmesi, e-posta ile gönderilmesi veya sistemdeki tespit edilen izinsiz girişlerden korunması gibi çeşitli yollarla uyarı veren bir yazılım, donanım veya bunların melez bir kullanımı olabilir. IDS olarak adlandırılan saldırı tespit sistemi, bir çeşit alarm sistemi olarak düşünülebilir. Bir IDS birkaç bileşenden oluşur (Marinova-Boncheva, 2007):

- a) **Sensörler (Ajanlar):** Güvenlik olayları oluşturur.
- b) **Monitör:** Olayları, uyarıları ve kontrol sensörlerini izlemek için kullanılır.
- c) **Merkezi Motor:** Veritabanındaki sensörler tarafından kaydedilen logları tutar ve bir güvenlik olayı oluştuğunda uyarı oluşturmak için kural sistemi kullanır.

Saldırı Önleme Sistemi (IPS), IDS'nin bir uzantısı olarak görülebilir; IPS, saldırının etkisiz hale getirilmesi için saldırının IDS tarafından aktif olarak alınmasını sağlar. Bu iki sistem genellikle birlikte kullanılır ve Saldırı Tespit ve Önleme Sistemleri olarak adlandırılır. IDS'ler, bilgi sistemi altyapısının güvenliğini artırmak için sürekli gelişen sistemlerdir. Ancak güvenlik kavramında hiçbir zaman doğal olmayan bir varsayım haline gelemeyen, saldırı tespit sistemleri dağıtım ve tespit olmak üzere iki yaklaşıma ayrılır. Şekil 3.3, savunma mekanizmasının nasıl çalıştığını göstermektedir (Inayat, Gani, Anuar, Anwar, & Khan, 2017).



Şekil 3.3. IDS'nin savunma mekanizmasının yaşam döngüsü

3.2.2.1. Dağıtım ortamına göre saldırı tespitin sınıflandırılması

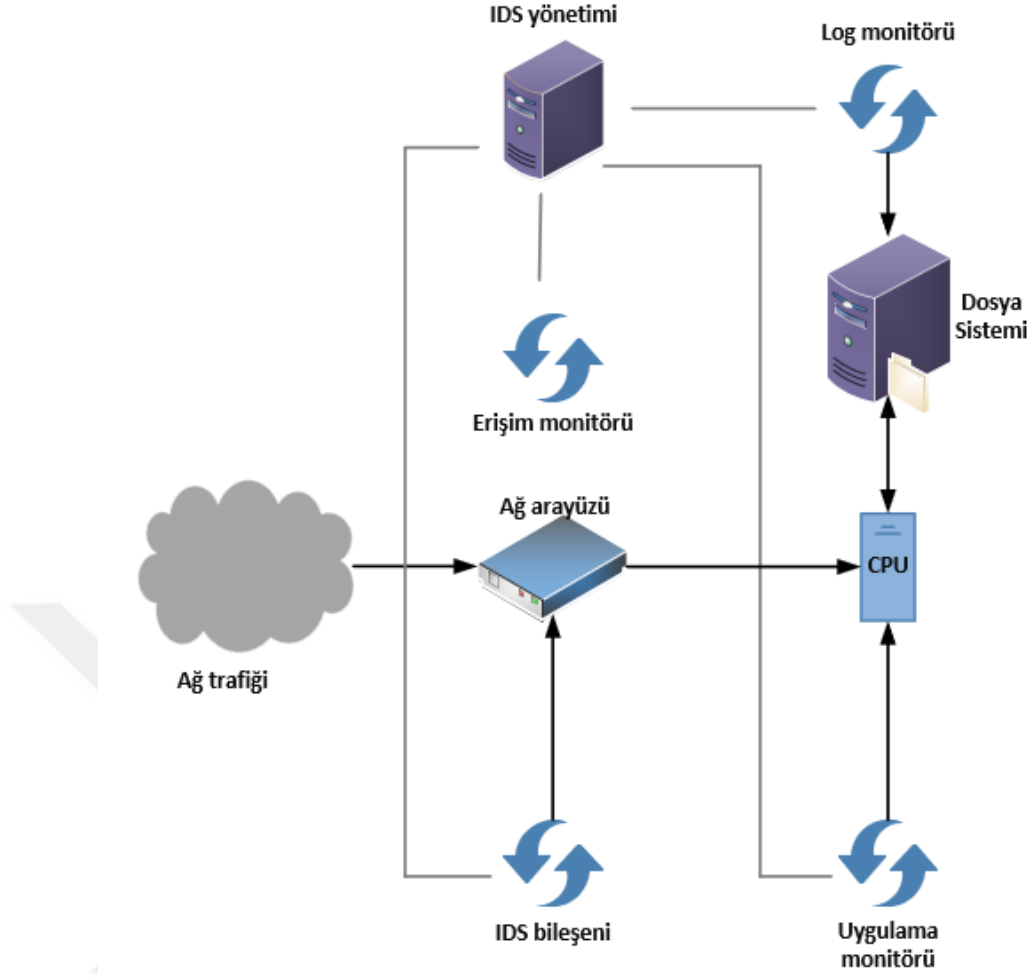
Saldırı tespit sistemleri, bilgi güvenliği fikrinin erken aşamalarında, kavram saldırısının ortaya çıkmasıyla görünür olmuştur. Saldırıları otomatik olarak tespit etmek için bir mekanizmaya duyulan ihtiyaç, sistem güvenliği için önemli bir unsur olacaktır. İlk aşamadaki başarılı sonuçlar, yeni bir teknoloji ve bilgi altyapısında köklü bir değişiklik olmadığı sürece, geliştirilmesi ve kullanılması gereken bir sistem olmaya devam ettiğini göstermektedir. Günümüzde, çeşitli algoritmalar kullanarak geliştirmekte olan sistemler, çok sayıda saldırıyı önleyerek ve saldırganları yavaşlatarak bilgi güvenliğinin temel taşlarını oluşturmaktadır. Ama bu sistemler tek başlarına sistemlerin sorunsuz çalışması için yeterli değildir.

Bir IDS, gerçek zamanlı olarak yanıt verebiliyorsa veya bir saldırı tespit edildiğinde yanıtın yakınındaysa aktif olarak kabul edilir. Ama gelen saldırıyı kaydeder ve daha sonra gözden geçirilmek üzere saklarsa pasif olur. Atakların tespiti, tetikleme mekanizmaları ile gerçekleştirilir. Saldırı tespitinde, sistemin ve ağ kaynaklarının yanlış veya olağandışı kullanım bilgileri bilinmelidir. Korunan ortamın kapsamına bağlı olarak çeşitli IDS yaklaşımları (Sunucu tabanlı, Ağ tabanlı, Kablosuz Ağ tabanlı, VMM tabanlı,

Bulut tabanlı vb.) kullanılabilir. Ancak, temel olarak, iki sınıf altında toplanırlar; Sunucu tabanlı saldırı tespit sistemleri ve Ağ tabanlı saldırı tespit Sistemleri.

3.2.2.1.1. Sunucu tabanlı saldırı tespit Sistemleri

Sonucu tabanlı saldırı Tespit sistemleri (Host-based Intrusion Detection Systems - HIDS), harici etkileşimleri seyrek olan anaçatı bilgisayarların ana hedefi olacak şekilde tasarlanmış ilk tip saldırı tespit yazılımıdır ("Trusted Computing Group," 2014). HIDS, bağımsız bir bilgisayardan toplanan bilgilerle çalışır. HIDS yalnızca bilgisayar sisteminden gelen ve giden paketleri izler ve şüpheli etkinlik 1 olarak algılanırsa kullanıcıyı veya yöneticiyi uyarır. Özellikle önemli sunucu sistemlerindeki gizli ve kritik bilgileri korumak için kullanılır. HIDS, aslında belirli bir makineye yapılan saldırıları önlemek için sunucularda veya iş istasyonlarında bulunan algılayıcılardır. HIDS, sistem durumunu izlemek için işletim sistemi denetim yollarını ve sistem log'u kayıtlarını kullanarak karar verebilir. Hangi kaynakların hangi programlara erişimi olduğunu tespit edebilir. Şekil 3.4'te sunucu tabanlı bir örnek gösterilmektedir (U Oktay & Sahingoz, 2013).



Şekil 3.4. Host-tabanlı IDS Mimarisine Genel Bakış

HIDS kullanmanın yararlarına ek olarak, bazı dezavantajları da vardır. Her sunucu için elde edilen bilgiler yapılandırılmalı ve yönetilmelidir. HIDS leri bu nedenle yönetmek zordur. Bilgi kaynakları ve HIDS için analiz motorları saldırıların hedefi olan sunucu bilgisayarda bulunduğundan, HIDS saldırılara maruz kalabilir ve saldırının bir parçası olarak devre dışı bırakılabilir. Ayrıca, izleme sırasında bilgisayar kaynaklarını kullanır, bu nedenle izlenen sistemde performans maliyetine neden olur. Diğer IDS'lerde olduğu gibi, HIDS'te de çeşitli tahsis edilmiş varlıklar vardır.

Sensör veya **ajan**: bir sunucu, iş istasyonu ve uygulama servisi gibi bir sunucuya veya yakınına yerleştirilmiş olurlar. Olay verileri; olayları kaydetmek ve diğer olası olaylarla bağlantı kurmak için kayıt servisine gönderilir. HIDS ajanları sayısız sunucu tipinde çalışabilir. HIDS sensörleri; sunucuları, kullanıcı bilgisayarlarını ve uygulama sunucularını izleyebilmektedir.

Sunucu: kullanıcılara genellikle web, e-posta veya FTP sunucularına veri göndermek veya almak için bağlı oldukları belirtilen hizmetleri sunmaya adanmış bir bilgisayardır.

Kullanıcı makinesi: bir kullanıcının diğer makinelere bağlanmasına izin veren masaüstü veya dizüstü bilgisayar gibi bir iş istasyonudur.

Uygulama servisi: web servisi veya veritabanı uygulaması gibi herhangi bir sunucu üzerinde çalışan bir yazılımdır.

Her kullanıcı makinesi farklı bir işletim sistemi çalıştırdığı veya bir servisi çalıştırdığı için, makineleri etkileyen saldırı türleri aynı zamanda bu makinelere özeldir. HIDS sensörü yalnızca ağ trafiğini değil, aynı zamanda kullanıcı makinelerini de izlediğinden, ajan, yazılımın bir parçası olarak kullanıcı makinelerinde bulunmaktadır. Mantıksal olarak ajan varlık, bir ağ cihazı olmak yerine, kullanıcı makine ve dış ağ arasında, trafiğin hizmet vermek için geçmesi gereken bir yazılım katmanı olarak görev yapmaktadır. HIDS nin hedef odağında belirli olaylar arasında, kötü amaçlı kodun ve bellek taşmasının yürütülmesi, bütünlük ve erişim de dâhil olmak üzere dosya sisteminin izlenmesi, kullanıcı kayıtları gözden geçirildiğinde gerçekleşen kayıt analizi ve son olarak ağ kurulum yapılandırmalarında yapılan değişiklikler gibi kod analizleri yer almaktadır.

3.2.2.1.2. Ağ tabanlı saldırı tespit sistemleri

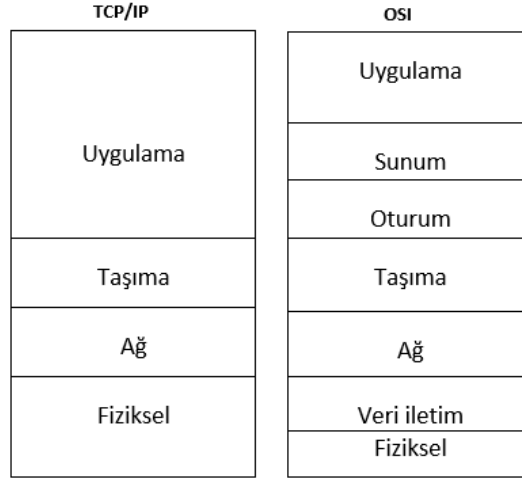
Ağ tabanlı IDS (Network based IDS - NIDS), belirli bileşenleri önceden tanımlanmış ve belirlenmiş ağ trafiği segmentlerine göre gözlemler, inceler ve analiz eder. Belirli bir sunucu değil, ağın kendisine odaklanır. Ağ üzerinden geçen trafiği veri kaynağı olarak görüntüler. NIDS, ağ segmentini dinleyerek veya cihazı değiştirerek bu ağ segmentine bağlı birden fazla sunucu etkileyen ağ trafiğini izleyebilir.

Genel olarak, ağ kartı karışık moddayken, içinden geçen tüm trafik yakalanır. Saldırıları ağ paketlerini yakalayarak ve analiz ederek tespit edilir. Trafik analizi yapmak ve istenmeyen ya da zararlı olay ve etkinlikleri aramak için ağa bir NIDS konulur. Ağ trafiği birkaç katman üzerine kuruludur, her katman bir katmandan diğerine veri aktarır.

Açık Sistemler Bağlantısı (Open Systems Interconnection - OSI) modeli ve İletim Kontrol Protokolü (Transmission Control Protocol TCP)/IP modeli, bu katmanların Şekil 3.5'te gösterildiği gibi nasıl organize edildiğini göstermektedir. OSI ve TCP / IP referans modellerinin katmanlı yapısı, TCP / IP modelinin kontrol katmanı ve altta donanımın

fiziksel adreslerinin kontrolü gibi verilerin kablo üzerinde nasıl iletildiğini açıklar. İnternet katmanı yönlendirmeyi kontrol eder ve IP yığını içerir.

İletişim katmanı veri akışını ve veri bütünlüğünü kontrol eder ve İletim Kontrol Protokolü'nü (TCP) ve Kullanıcı Datagram Protokolünü (User Datagram Protocol -UDP) içerir. Son bölüm, programların kullanımına en aşına olan en karmaşık uygulama katmanını içerir. Uygulama katmanı trafiği, Hiper Metin Aktarım Protokolü (Hypertext Transfer Protocol - HTTP), FTP ve Basit Posta Aktarım Protokolü (Simple Mail Transfer Protocol - SMTP) gibi protokolleri içerir. Çoğu NIDS her katmanda meydana gelen istenmeyen trafiği tespit edebilmesine rağmen, bu trafiğin çoğu uygulama katmanında yer almaktadır. IP katmanında istenmeyen trafiğin tespiti için güvenlik duvarları kullanılır ve istenmeyen trafiğin tespiti tek bir varlıkta aşırı yüklenmez ve işbölümü gerçekleştirilir.



Şekil 3.5. TCP/IP ve OSI mimarisi

Sensörler: sensörler veya araçlar, ağ trafiğini izleyen ve zararlı olup olmadığına karar verenlerdir. Çoklu sensörler genellikle ağ içindeki belirlenmiş noktalara yerleştirilir. Ağ bağlantıları güvenlik duvarlarına, anahtarlama cihazlarına, yönlendiricilere veya ağı bölen diğer noktalara yerleştirilir.

Veritabanı Sunucusu: NIDS'in depolama bileşenleridir. Sensörlerden gelen olaylar ve bu olaylarla ilgili veriler bu sunuculara kaydedilir. Veritabanları yüksek depolama alanına sahip olmalı ve iyi performans göstermelidir

Yönetim Sunucusu: Analiz sonuçlarının tüm sensörlerden gönderilebileceği merkezi bir noktada bulunur. Yönetim sunucuları, güvenlik nedenlerinden ötürü gerçek ağdan bağımsızdır. Genellikle bir ağ üzerinden algılayıcılarla iletişim kurar ve algılayıcı

raporlarına dayanarak kararlar alır. Farklı sensörlerden gelen veriler ağı farklı bölümlerinde meydana gelen trafikle ilişkilendirilebilir.

Konsol: NIDS için kullanıcı arayüzüdür. Bunlar, NIDS'in yönetici tarafından kontrol edilmesini, yapılandırılmasını ve izlenmesini sağlayan bölümlerdir. Konsol, yerel olarak çalışan bir yazılımdır veya güvenli bir web portalı olabilir. Bileşenler arasındaki trafik güvenli bir şekilde gerçekleştirilmelidir. Hiçbir bileşen bu trafiği değiştiremez veya görüntüleyemez. Engellenen trafik, saldırganın saldırı trafiğini değiştirmesine neden olabilir. NIDS'te, sensörler ağ üzerinde farklı şekillerde konumlandırılabilir.

Inline Konumlandırma: Ağı ikiye bölen yönlendirici veya güvenlik duvarı gibi iki cihaz arasına yerleştirilir. Bu, iki cihaz arasındaki trafiğin, sensörün trafiği analiz edebileceğini garanti ederek sensörden geçtiği anlamına gelmektedir. Bu şekilde yerleştirilen bir sensör, zararlı olarak kabul edilen trafiği önlemek için kullanılabilir. Bu tür sensörler analiz edilecek daha az trafik sağlamak için güvenlik duvarının arkasına ve iç ağına girişine yerleştirilmektedir.

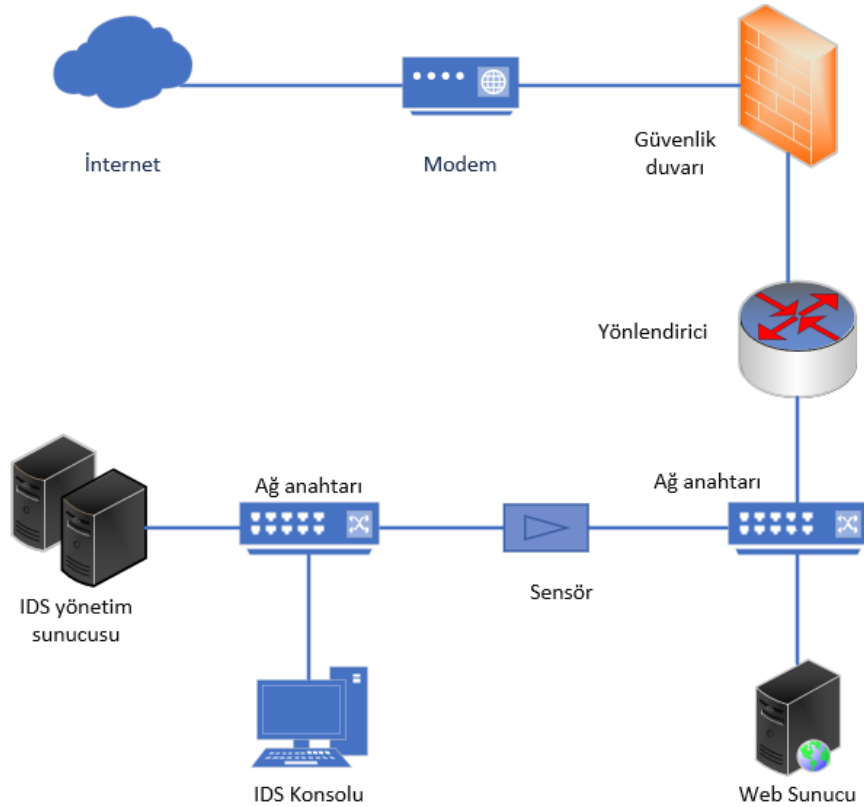
Pasif Konumlandırma: Pasif bir sensör, ağdan geçen trafiği kendi üzerine kopyalayarak analiz yapar. Bu kopyalanan trafiğin birden fazla kaynağı olabilir.

Yayılma Portu Konumlandırma: Bir anahtarlama cihazından geçen tüm trafik tek bir porta kopyalanır. Tüm trafiği düşük miktarda ağ trafiğinde gözlemlemenin kolay bir yolu olsa da, anahtarlama aygıtı ağ trafiği arttıkça tüm trafiği kopyalayamayabilir.

Ağ Musluğu Konumlandırma: Bu tür sensörler, fiziksel katmandan geçen ve genellikle fiber optik kablolardaki tüm trafiği algılar ve sinyal seviyesini düşürmeden çoğaltılır. Doğrudan aktarım ortamına yerleştirilirse, tüm bağlantının kesilmesine neden olabilirler. NIDS tarafından tespit edilebilecek bazı olaylar örneklenmiştir. Sadece keşif amacıyla yapılan olaylar tehlikeli değildir, ancak tehlikeli saldırılara neden olabilir. Bu tür olaylar ağ katmanında port taraması olarak gerçekleşir. Sistemlerde çalışan servisler izin verilen bağlantıların yapılması için bazı portlar açar. Bir port taraması sırasında, saldırgan her porttan bir bağlantı kurmaya ve hangi servislerin çalıştığını belirlemeye çalışır. Bu şekilde, işletim sistemi ve bir web sunucusunda çalışan sürüm hakkında bilgi sahibi olabilirsiniz. NIDS ayrıca ağ, iletişim ve uygulama katmanındaki saldırıları tespit edebilir. Bu tür saldırılar, Hizmet Reddi (DoS) saldırıları ve bilgi hırsızlığında kullanılan kötü amaçlı kodları içerebilir. NIDS ayrıca daha az tehlikeli ancak istenmeyen trafiği tespit etmek için de kullanılabilir. Böylece, arka kapılar ve politika ihlalleri gibi istenmeyen hizmetler önlenir.

NIDS temel olarak ağdan geçen paketlerle ilgilenmektedir. Ağdaki belirli noktalarda bulunan sensörlerden geçen veri, sistemdeki mevcut imzalarla karşılaştırılarak analiz edilmektedir. Giriş katman filtresi, hangi paketlerin kabul edileceğini ve hangilerinin atılacağını veya paketin saldırı tanıma modülüne gönderilip gönderilmeyeceğini belirler. Bir saldırı tespit edilirse, yanıt modülü saldırıya yanıt olarak alarm oluşturma mekanizmasını tetikler. Güvenlik için sensör ve monitör arasındaki trafiği şifrelemek veya ayrı bir ağa sensörleri ve monitörleri dahil etmek önemlidir. Bilgili ve deneyimli bir saldırgan için, sensör ve monitör arasındaki trafik (alarmlar, durum kayıtları, diğer paketler vb.) ağa saldırmak için önemli bilgiler içerir. Sensörler ve monitörler Servis Reddi (DoS) saldırılarını önlemek için farklı bir ağda bulunmaktadır.

NIDS iki bileşenden oluşur: araçlar ve yazılımlar. Araç; bir işletim sistemi, uygulama ve ağ kartından oluşur. Yazılım; IDS yazılımı ve bazen kullanıcı tarafından kullanılan işletim sisteminden oluşur. NIDS araç tabanlı sistemlere kıyasla yalnızca yazılım olarak genellikle daha az maliyetli olsa da, donanımla çalışmak üzere ayrı ayrı yapılandırılmaları gerekir. NIDS genellikle bir cihazın parçası değildir, ancak çok ayrı bir donanım içeren ayrı bir fiziksel cihaz olabilir. Şekil 3.6 (U Oktay & Sahingoz, 2013) bir NIDS örneği göstermektedir.



Şekil 3.6. Ağ tabanlı IDS mimarisine genel bakış

3.2.2.1.3. Kablosuz tabanlı saldırı tespiti

Kablosuz teknolojilerin çok popüler olması, kablosuz iletişimin doğası ve ağlar arasındaki sınırlar nedeniyle saldırı tespit ve önleme yaklaşımları önemli bir çalışma alanı haline gelmiştir. Kablosuz IDS'ler yapı olarak NIDS'lere benzer. Bunun nedeni, kablosuz ağların kablolu ağlarla benzer saldırılara maruz kalmasıdır. Kablosuz ağları geleneksel kablolu ağlardan ayıran işlevsellik ve zayıflık nedeniyle, kablosuz bir IDS kablosuz ağ saldırılarını normal ağ saldırılarında olduğu gibi izleyebilir. Kablosuz IDS sensörleri, tüm kablosuz trafiği yönlendirme olmadan izleyebilen bağımsız cihazlardır ve kablosuz ağlarda ve kablolu ağlara bağlanıldığında erişim noktaları olarak işlev görebilir. Kablosuz bir LAN sensörünün konumu, sensörün fiziksel konum aralığını doğrudan etkiler. Bir sensör yalnızca kablosuz LAN'daki cihazları değil, kablosuz ağa bağlanabilme yeteneğine sahip cihazları da izleyebilir. Bir kablosuz IDS, tıpkı geleneksel bir NIDS gibi bir sensöre, bir yönetim sunucusuna, bir kayıt veritabanına ve bir konsola sahiptir. Kablosuz sensörler, kablolu ağlarda görüntülenebilen ve kablosuz ağlara özgü olaylar gibi birçok olayı izleyebilmektedir ve takip edebilmektedir. Yetkisiz LAN ve kablosuz cihazlar, düşük güvenli kablosuz cihazlar, sıradışı veri yolları, kablosuz tarayıcılar, DoS saldırıları ve izinsiz giriş saldırıları gibi anormallikleri algılayabilir.

3.2.2.1.4. Ağ davranışı anomalisi tespiti

Ağ Davranışı Anomalisi Tespiti (Network Behavior Anomaly Detection -NBAD), ağ trafiğinin resmi ya da istatistiksel durumundan ayrı paketleri düşünmeden trafiğin kötü amaçlı olup olmadığını belirlemenizi sağlayan bir saldırı algılama yöntemidir. NBAD sensörleri, anahtarlama cihazları, DMZ'ler ve trafiğin farklı bölümlere ayrıldığı ağ üzerindeki anahtar noktalara kurulur. Sensörler, görev bölgelerinde hangi tür trafik türünün bulunduğunu ve ardından DoS saldırılarını algılar, ağda tarama yapar, solucanlar, beklenmeyen uygulama hizmetleri ve kural ihlallerini algılar. Geleneksel kablolu NIDS ve NBAD sistemleri, sensörler ve yönetim konsolları gibi sahip oldukları yapı içeriğinin bir kısmını paylaşır ve geleneksel NIDS'lerin aksine, NBAD sistemlerinin çoğu zaman veritabanı sunucuları yoktur. NBAD sistemleri, trafik taban çizgisinden ayrıldığında karar vermek için çalışır. Bu genellikle DoS saldırılarını ve solucanlarını tespit etmede

yararlıdır. Diğer IDS'lerde olduğu gibi, NBAD sistemleri, trafik akışını durdurarak kötü amaçlı trafiğin durdurulmasının önlenmesinde yararlı olabilir.

3.2.2.2. Tespit yöntemlerine göre saldırı tespiti

Algılamaya göre IDS üç kategoriye ayrılır: İmza tabanlı, Anomali tabanlı ve Bilgi durum analizi tabanlı. Her bir kategori aşağıdaki alt bölümlerde gözden geçirilecektir.

3.2.2.2.1. İmza tabanlı IDS'ler

İmza tabanlı IDS'ler bilinen saldırı türlerine göre önceden tanımlanmış kurallarla yapılan saldırı girişimlerini tespit eder ve yeni saldırı türlerine karşı başarılı değildir. Bu tespit mekanizmaları, olayları önceden tanımlanmış kurallarla karşılaştırdıkları için uygulama ve yönetim açısından uygulanması en kolay olanıdır. Daha önce saldırıya uğramış bir kaynağı kayıtlarında tutamaz. Bu nedenle, imza tabanlı izinsiz giriş tespiti, daha önce saldırıya uğramış bir kaynaktan meydana gelebilecek karmaşık bir saldırı için risk oluşturur.

İmza bazlı tespitler bilinen saldırı türlerine karşı başarılı olmasına rağmen, yeni saldırı türlerine karşı etkili değildir. İmza tabanlı tespitler, paket ve kayıt girişleri gibi mevcut durumları "string" ifadeleri olarak oluşturulan imzalarla karşılaştırdıkları için fizibilite açısından en basit izinsiz giriş tespit yöntemleridir. Bu yöntem pek çok ağ ve uygulama protokolüne aşina değildir, çünkü karmaşık iletişimlerini takip edememektedir. Örneğin, web sayfaları tarafından "403" durum koduna sahip bir sunucu tarafından reddedilme isteği imza tabanlı algılama yöntemleriyle anlaşılmamaktadır.

Ek olarak, mevcut bir talebi değerlendirirken, önceki talepler hakkında bilgisi yoktur. Bu kısıtlamalar nedeniyle, imza tabanlı izinsiz giriş tespit yöntemleri, birden fazla durum bilgisini birleştirerek karmaşık saldırıları tespit edemez.

3.2.2.2.2. Anomali tabanlı IDS'ler

Anormalliğe dayalı IDS'ler, bir sistemin davranışını gözlemleyerek bu davranışları analiz etmektedir. Anomaliye dayalı tespit teknikleri imza bazlı tekniklerle birleştirilirse, saldırı tespitindeki başarı oranları artar. Genellikle çoklu görev veya ağ erişimi gerçekleştirerek bilgi kaynaklarını tüketen yeni geliştirilen kötü amaçlı yazılımlar

gibi bilinmeyen saldırı türlerine karşı tercih edilir, bu nedenle sistemin olağan davranışından farklı belirlemeler yapabilir. Anormallik tabanlı izinsiz giriş tespiti, normal olayların tanımlarını mevcut durumdan belirgin sapmalarla karşılaştırır. Bir kullanıcının, sunucunun, ağ bağlantısının veya uygulamanın profilleri, belirli bir aktivitenin özellikleri belirli bir zaman diliminde gözlemlenerek oluşturulur. Örneğin; Bir ağ profili, ağ bant genişliğinin %13'ünün bir iş günü mesai saatleri içinde web işlemleri tarafından üretildiğini gösterebilir. Bir IDS, mevcut bir faaliyet özelliğinde bulunan sınır değerlere göre profilin normalliğini karşılaştıran istatistiksel bir yöntem kullanır. Normlardaki sapma örnekleri, bir kullanıcı tarafından gönderilen e-posta sayısını, bir sunucuda başarısız erişim girişimlerini ve bir süre içinde bir sunucuda işlemcilerin kullanımını içermektedir.

Anormalliğe dayalı izinsiz giriş tespit yöntemlerinin en büyük yararı, önceden bilinmeyen tehditlere karşı oldukça başarılı olmalarıdır. Yeni geliştirilen bir kötü amaçlı yazılımın virüslü bir bilgisayar olduğunu düşünürsek, kötü amaçlı yazılımlar bilgisayar kaynaklarını tüketmeye başladığında, çok fazla e-posta gönderirken veya ağ bağlantılarını normalden fazla açarken kolayca algılanabilir.

3.2.2.2.3. Bilgi durum analizi tabanlı IDS'ler

Bilgi durum analizi tabanlı IDS'ler bilinen protokollerin özelliklerini kullanır. Anormalliğe dayalı yöntemlerden farklı olarak, protokollerin yapısını tanımlarlar ve bu nedenle imza temelli IDS'lere benzerler. Yaygın olarak kullanılan protokollerin önceden belirlenmiş tanımlarına dayanan ve rutin aktivitelerdeki sapmalara dayanan bir yöntemdir. Protokolün geliştiricileri tarafından evrensel olarak tanımlanan özelliklere uygun olarak bu protokollerin genel davranışlarından ve eylemlerinden oluşan profilleri kullanır.

'Durum' sözcüğü, protokolün tasarımından dolayı şu anki durumunun ne olması gerektiğini anlayabildiği anlamına gelir. Bir kullanıcı bir dosya aktarım protokolü (File Transfer Protocol - FTP) oturumu başlattığında, bu oturum ilk önce kimlik doğrulaması olmadan başlar. Bu durumda, kimliği doğrulanmamış kullanıcı yardım bilgilerini görüntüleme gibi yalnızca birkaç temel komut çalıştırabilir. Durumu anlamamanın en önemli noktalarından biri istek ve cevapların eşleşmesidir. Böylece bir FTP kimlik doğrulama talebi geldiğinde, IDS gelen talepte durum kodunu bulabilmektedir. Kullanıcı başarıyla kimliği doğrulandıktan sonra, oturum kimliği doğrulanır ve kullanıcının yetkisi

dâhilindeki komutlar çalıştırılabilir. Kimliği doğrulanmamışken, bu komutların çoğu yürütüldüğünde şüpheli olarak kabul edilecektir. Ancak, bu komutlar doğrulanmış bir oturumda yürütüldüğünde normal aktivite olarak değerlendirilirler.

3.2.2.2.4. Yanlış pozitif ve yanlış negatif

IDS'ler her zaman sistem yöneticilerine kesinlikle doğru bilgiler sağlamamaktadır. Sistemdeki bazı olağandışı hareketler izinsiz girişimleri olarak algılanabilir ve aslında sistem tarafından gerçekleştirilen gerekli eylemler olabilir. IDS'ler yanlış pozitif veya negatif tespitler nedeniyle çok iyi yapılandırılmalıdır. Ağ trafiğinin karmaşıklığı nedeniyle, bir IDS'nin hatasız ve yanılgı olmadan çalışması mümkün değildir. Hatalı sonuçlar bağlamında, IDS tespitleri iki gruba ayrılır: yanlış pozitif ve yanlış negatif. Yanlış pozitifler, IDS'nin şu anki durumu bir saldırı olmasa da olayın ve trafiğin doğası gereği bir saldırı olarak algılamasıdır. Yanlış negatifler, IDS'nin trafiği ve aslında bir saldırı girişiminde bulunan olayları tespit edememesidir. Her iki durumda da, sistem güvenliği yöneticilerinin sistemde kalibrasyon ve gerekli yapılandırma değişikliklerini yapması gerekmektedir. Yanlış pozitiflerin yüksek oranlarda tespit edilmesi, genellikle sistemin güvenliği için bir sorun değildir ve kabul edilebilir sayılmaktadır. Yalnızca sistem güvenliği yöneticilerini gereksiz yere meşgul eder. Algılanamayan yanlış negatifler, sistem güvenliği yöneticilerine herhangi bir yük getirmez, ancak sistemin güvenliği için büyük tehlike oluşturmaktadır.

3.2.2.3. Güvenilir Bilişim - Trusted computing

Güvenilir Bilişim (Trusted Computing - TC), Trusted Computing Group (TCG) tarafından geliştirilen ve desteklenen bir metodolojidir ("Trusted Computing Group," 2014). Güvenilir bilişim veya bilgi işleme, bilgisayar güvenliği sorunlarını donanım geliştirmeleri ve ilgili yazılım değişiklikleri yoluyla çözmek için teknoloji ve tavsiyeler sunan geniş bir konsepttir. Bilgi sistemlerini oluşturan bileşenlerin her biri arasında gizlilik, bütünlük, erişilebilirlik ve kurtarılabilirlik ihtiyacını temel alan bir 'güven ilişkisi' planlamaktadır. Birçok büyük donanım üreticisi ve yazılım sağlayıcısı, TCG ile işbirliği yapmaktadır ("Trusted Computing Group," 2014). TCG, bilgisayar kaynaklarını kötü niyetli kişilerin son kullanıcı haklarını ihlal etmeden tehdit ettiği tehditlere karşı koruma özellikleri sunmaktadır.

Güvenilir Platform Modülü (Trusted Platform Module - TPM), bilgisayar kaynaklarını yetkisiz değişikliklere ve tehditlere karşı korumak için TC yaklaşımının ana parçası olarak kullanılır. TPM, anakart üzerindeki entegre bir devredir ve sistemde çalışan yazılım tarafından iyi tanımlanmış olan veya olmayan komutların ve örneklerin bütünlüğünü kontrol etmektedir (Kong, 2010).

TPM, şifreleme anahtarlarına dayalı güvenlikle ilgili temel işlemleri sağlamak için tasarlanmıştır. İşletim durumu platformu hakkındaki bilgileri, TPM'deki platform yapılandırma kayıtlarında (Platform Configuration Registers - PCRs) depolar. Mevcut platformdan platform isteklerini doğrular. TC, uzaktan onaylama, mühürlü depolama, güvenilir önyükleme gibi teknolojiler içermektedir. Kullanıcılar, uzaktan doğrulama ile platformda çalışan yazılımdan sistemle ilgili raporlar alabilmektedir. Bu rapor, TPM tarafından imzalanan ve onaylanan PCRs yapılandırma değerlerinin bir listesidir (Kong, 2011). TPM'li bilgisayarlar, şifreleme anahtarları oluşturabilir ve bunları yalnızca TPM tarafından çözülebilmeleri için şifreleyebilir. Yaygın olarak "kaydırma" veya "bağlama" olarak bilinen bu işlem, anahtar gösterilmesini engelleyebilir. Her TPM, TPM'de depolanan saklama kök anahtarı adı verilen bir ana "kaydırma" anahtarına sahiptir. TPM'de oluşturulan anahtarın özel kısmı hiçbir zaman başka bir bileşene, işleme, yazılıma veya kişiye gösterilmez.

3.2.3. Bulut bilişime yapılan saldırılar

Bulut bilişim sistemlerine yapılan saldırılar, geleneksel IT sistemlerine yapılan tüm saldırıları kapsar, ancak ağırlıklı olarak bulutta yer alan bazı saldırı türleri vardır.

3.2.3.1. Bulutun içerisinden gelen saldırılar

Yetkili bulut kullanıcıları, sistem yöneticileri veya bulut operatörleri, aslında kullanıcıların gerçekleştirdiği davranışlarda yetki sahibi olmamalıdır. Bu kullanıcılar, bulut kullanıcılarının bilgilerine yetkilerinin ötesinde erişir, bu bilgileri amaçlarına göre değiştirir ve bunları bulut kullanıcılarına zarar verecek şekilde kullanırlar.

3.2.3.2. Flooding saldırıları

Bu tür bir saldırıda, saldırganlar zombiler adı verilen yakalanan ağ kaynakları aracılığıyla ağ üzerinden mağdurlara çok sayıda paket göndermektedir. Paketler TCP, UDP, İnternet Kontrol Mesaj Protokolü (ICMP) veya bunların bir kombinasyonu olabilir. Bu saldırılar genellikle yetkisiz ağ bağlantıları üzerinden gerçekleştirilir. Bulut bilişimin doğası gereği, Sanal Makineler (VM) deki tüm ağ bağlantıları internet üzerinden yapılır. Bu nedenle, bulut kullanıcılarının, zombi kaynakları tarafından DoS veya Dağıtılmış Hizmet Reddi (DDoS) saldırılarına maruz kalması kaçınılmazdır.

Flood saldırıları yetkili kullanıcıların servis kullanılabilirliğini etkiler. Tek bir hizmet sağlayan bir sunucudaki saldırgan, sunucuya saldırabilir ve bu hizmete erişimi engelleyebilir. Bu tür saldırılara DoS saldırıları denir. Sunucu kaynakları flood saldırılarının bir sonucu olarak hareket etmek zorunda kalır ve sunucudaki diğer hizmetlerin çalışmasını engellerse, bu saldırılara dolaylı DoS saldırıları denir. Yüksek miktarda donanım kullanan flooding saldırıları ve bulut hesaplama, kullandıkça öde modeliyle ücretlendirilir, bulut kullanıcılarının masrafları gereksiz yere artmaktadır.

3.2.3.3. Yetkili kullanıcı saldırıları

Yetkili bir kullanıcının hesap ve şifre bilgilerini alan bir saldırgan sisteme sınırsız erişime sahiptir. Arabellek taşmaları, yönetici ayrıcalıklarıyla çalışan işlemler için konsol bağlantıları oluşturmak için kullanılır. Bu tür saldırılar, bir uygulamanın statik olarak tanımlanan tampon belleğine kapasitesinden daha fazla bilgi istemek suretiyle gerçekleştirilir ve tampon belleğe uymayan bilgiler saldırganlar tarafından ele geçirilir. Yetkili bir kullanıcının hesap ve şifre bilgilerini alan bir saldırgan, sunucuya ve dolayısıyla sunucuda çalışan tüm VM'lere erişebilmektedir.

3.2.3.4. Port tarama

Port tarama, sistemdeki açık, kapalı ve filtrelenmiş portların algılanmasını sağlayan bir saldırıdır. Port tarama ile saldırganlar, açık portlar üzerinden bir sistemde çalışan servisler ve ağ bağlantıları için IP adresi, ortam erişim kontrolü (MAC) adresi, yönlendirici, ağ geçidi ve güvenlik duvarı kuralları gibi ayrıntılı bilgiler alabilirler. TCP taraması, UDP taraması, SYN / FIN / ACK taraması ve Pencere taraması en sık kullanılan bağlantı noktası tarama saldırılarıdır. Port taraması, kendi başına kullanılan bir saldırı olmamasına rağmen, bulut bilişim kapsamındaki bir saldırgan, port taraması yoluyla

hizmet olarak sunulan uygulamaların açık bağlantı noktalarını algılar ve elde edilen bilgilere göre gerçek saldırıyı gerçekleştirir.

3.2.3.5. Sanal Makineye veya hiper yöneticiye yapılan saldırılar

Hipervizörün kazanılmasından sonra, sanal ortamda çalışan tüm sanal makinelerin kontrolü ele alınmaktadır. Sıfır gün saldırıları, saldırganların sanal makinelere saldırmak ve sanal ortamda çalışan hiper denetleyiciyi veya diğer sanal makineleri yakalamak için kullandıkları yöntemlerden biridir. Bu, mevcut bir güvenlik açığını yamalamadan veya güncellemeden önce bu güvenlik açığını kullanarak sistem ve yazılım geliştiricileri tarafından gerçekleştirilir.

3.2.3.6. Arka kapı kanalı saldırıları

Saldırganların buluttaki bir düğümü yakalamasını ve DDoS saldırıları için onu bir zombi kaynağına dönüştürmesini sağlayan pasif bir saldırı türüdür. Truva atları ve sistemdeki benzer yapılar ele geçirildiğinde, sistem bir zombi olarak kullanılabileceği gibi kullanıcılara ait bilgilere de erişilebilir (Modi et al., 2013).

3.2.4. Sistem bileşenleri

3.2.4.1. Sensör

Sensör, bilgi toplamak için bir sistem veya ağ üzerinden yerleştirilmektedir. Birçok kaynaktan ağ paketleri dahil olmak üzere çeşitli bilgileri toplamakta, kayıt etmekte ve sistem çağrılarını takip etmektedir. Giriş verileri toplandıktan sonra düzenlenmekte ve bir veya daha fazla analizciye gönderilmektedir.

3.2.4.2. Analizci

Sensörlerden yönlendirilen verileri toplar ve bir saldırı girişimi olup olmadığını tespit eder. Analistin çıktısı, saldırı raporunda yer alan kanıtları desteklemektedir. Ayrıca sistem güvenlik yöneticilerine rehberlik ederek çıktı ve raporlar üretir.

3.2.4.3. Kullanıcı arayüzü

Bunlar son kullanıcılar ile sistem arasındaki bağlantıyı arayüz üzerinden sağlayan parçalardır, kullanıcılar sistemi kontrol edip yapılandırabilir ve sistem raporları oluşturabilirler.

3.2.4.4. Bal çanağı

Dağıtılmış bir IDS'de, saldırganlar için bir tuzak olarak uygulanır ve 'bal çanağı' olarak adlandırılır. Bal çanağı, sisteme yapılan saldırılar için erken uyarı mekanizmasıdır. Kritik sistemler önünde kurulduğu bir ortamda, yem görevi görerek saldırı kaynaklarının analizi kapsamında veri toplanmasını sağlamaktadır. Birçok IDS üreticisi sisteme araştırma amaçlı ballı çanak yerleştirir ve bunlardan gelen verileri yeni saldırı imzaları geliştirmek için kullanmaktadır.

3.3. Makine Öğrenmesi

Makine öğrenmesi (ML), 1959'da sayısal öğrenme ve yapay zeka model tanıma çalışmaları ile geliştirilen bir bilgisayar bilimi alt dalıdır. Makine öğrenme algoritmaları çok popülerdir ve tıp, bankacılık, güvenlik, otomotiv, eğlence, bilim ve pazarlama gibi birçok alanda kullanılmaktadır. Temel olarak, ML otomatik öğrenme ve gelişme ilkesine dayanmaktadır.

Makine öğrenimi, çeşitli algoritmalar ve yöntemlerle verilerdeki bazı kalıpları araştırır ve bu kalıplara karşılık gelen etiketlere bakarak öğrenir, daha sonra deneyimlerinden çıkartabilecek sistemler geliştirir (öğrendiklerine benzer bir durumla karşı karşıya kaldıklarında). Bu olasılık, çeşitli matematiksel ve istatistiksel yöntemler kullanarak birçok algoritma tarafından sağlanır. Bu yöntem ve algoritmalarından birini veya birkaçını kullanarak, modeller oluşturulur ve neyin en verimli, en kesin ve en hızlı şekilde tahmin edileceğini tahmin etme amaçlanır. Bilinen makine öğrenme yöntemleri aşağıda sunulmuştur:

3.3.1. Gözetimli öğrenme

Gözetimli öğrenmede bilgisayar problemi ile ilgili girdi verileri ve bu verilere karşılık gelen istenen çıktı verileri önceden bir uzman tarafından verilir. Amaç, giriş verileri ile çıktı verileri arasında bir ilişki inşa etmektir. Bu nedenle, problem için genel bir model oluşturulur.

Gözetimli öğrenme ikiye ayrılır: Sınıflandırma ve Regresyon. Sınıflandırmada, girişler iki veya daha fazla sınıfa ayrılır. Algoritma, daha önce görmediği uygun sınıflara veri atayabilmelidir. Bu süreç genellikle gözetimli öğretim yöntemleriyle yapılır. Regresyon, gözetimli bir öğrenme modelidir ve çıkışları sürekli'dir. En çok kullanılan gözetimli öğrenme algoritmaları; Naive Bayes, Lojistik regresyon, Karar Ağaçları (DTrees), K-En Yakın Komşu (KNN), Yapay sinir ağı ve Destek vektör makineleridir. Bazı algoritmalara aşağıda kısaca değinilmiştir:

3.3.1.1. Naive bayes

Naive Bayes, eğitim verilerinden hesaplanan her sınıf için önceden olasılığın bulunduğu Bayes teoreminin basit bir uygulamasını kullanan sınıflandırma algoritmasıdır. En yüksek olasılıkla sınıf için bir tahmin yapar ve ikili sınıflandırma ve çok sınıflı sınıflandırma problemlerini destekler. Naive Bayes'in çok etkili bir sınıflandırma algoritması olduğu gösterilmiştir.

3.3.1.2. Lojistik regresyon

Lojistik regresyon (LR), her girdi değeri için doğrusal olarak bir regresyon fonksiyonu ile birleştirilen ve lojistik fonksiyonla dönüştürülen bir katsayı öğrenen ikili sınıflandırma algoritmasıdır. LR hızlı ve basit bir tekniktir, ancak bazı problemlerde çok etkili olabilir.

3.3.1.3. Karar ağaçları

Karar ağaçları (DTrees), bir veri örneğini değerlendirmek için; bir ağaç oluşturup, ağacın kökünden başlayıp iyi bir tahmin yapıncaya kadar yaprakları aşağı taşıyarak çalışır. Bir karar ağacı oluşturma süreci, tahminler yapmak ve ağacın sabit bir derinliğe kadar süreci yinelemek için en iyi bölünme noktasını seçerek çalışır.

3.3.1.4. Yapay sinir ağı

Yapay sinir ağları, ilk olarak iki bilgi arasındaki ilişkiyi öğrenmek için kullanılan algoritmik tekniklerdir ve makul bir şekilde yeni girdi-çıkı çiftleri elde etmek için "genelleme" yapar. Sinir Ağlarının saldırı tespitinde kullanımının temel sebebi, veriyi (tamamlanmamış verilerden) genelleme ve verileri normal veya müdahaleci olarak sınıflandırılabilme yeteneğidir.

3.3.1.5. Destek vektör makinesi

Destek vektör makineleri (SVM), doğrusal ve doğrusal olmayan verilerin sınıflandırılması ve regresyon analizi için kullanılan desenleri tanımlayan verileri analiz eden, ilişkili öğrenme algoritmalarıyla bir gözetimli öğrenme modelidir. SVM, daha az örnek verinin bulunmasına rağmen, müdahaleleri tespit etmede kullanışlıdır. SVM yüksek boyutlu verilerle bile iyi genelleştirme yeteneğine sahiptir ve yapay sinir ağı tabanlı sınıflandırıcıya göre daha az sayıda eğitim örnekleri gerektirmektedir.

3.3.2. Gözetimsiz öğrenme

Gözetimsiz öğrenme, gizli yapıyı etiketsiz veriden tanımlamak için bir fonksiyon üreten makine öğrenme görevidir. En çok bilinen gözetimsiz öğrenme yöntemi kümelemedir. Kümeleme, giriş verilerinin gözetimsiz olarak gruplanması işlemidir. Sınıflamanın aksine gruplar önceden tanımlanmamıştır. En çok kullanılan gözetimsiz öğrenme algoritmaları K-En Yakın Komşu (KNN), K-Ortalama (K-mean), Beklenti maksimizasyonu (EM), DBSCAN, OPTICS, CURE, CLARA ve bazı diğer algoritmalar. Bu algoritmalarından bir kısmı aşağıda açıklanmıştır:

3.3.2.1. K-en yakın komşu (kNN)

KNN, güvenlik teknikleri de dâhil olmak üzere pek çok problem için etkin bir gözetimsiz öğrenme metodudur (Tavallaee, Bagheri, Lu, & Ghorbani, 2009). KNN, aynı özelliklere sahip elemanların kümelenmesine dayalıdır; kendisine yakın olan K komşusuna dayanan bir test örneğinin sınıf kategorisini kararlaştırır. KNN'deki K değeri, veri kümesinin boyutuna ve sınıflandırma problem türüne bağlıdır.

3.3.2.2. K-means

K-ortalama, bilinen kümeleme problemini çözen, en basit gözetimsiz öğrenme algoritmalarından biridir. Temel fikir, her küme için bir tane olmak üzere K merkezleri tanımlamaktır. Bu merkezler kurnaz bir şekilde yerleştirilmelidir çünkü farklı yer farklı sonuçlara neden olur. Her bir yeni satır verisi geldiğinde bu K adet merkeze mesafeler hesaplanarak atanma işlemi aynı veri kümesi noktaları ve en yakın yeni merkez arasında yapılmalıdır. Bu döngüsel bir işlemdir. Bu döngü sonucunda, K merkezlerin yerinin daha fazla değişiklik yapılamayınca kadar kademeli olarak değiştirildiği görülür.

3.3.2.3. Beklenti maksimizasyonu

Beklenti maksimizasyonu (EM), olasılık dağılımının θ parametresinin maksimum olasılık tahmincisini bulmak için bir yöntemdir. EM algoritması, bazı çok modlu verilerin dağılım parametreleri için en iyi hipotezi oluşturmak için kullanılabilir. Dağılım parametreleri için en iyi hipotez, maksimum olasılık hipotezidir (yani, olasılığı en yükseğe çıkaran hipotezdir). EM, makine öğrenimi ve bilgisayarlı görüntü işlemede veri kümelemesi için sıklıkla kullanılır. Tıp görüntülerinin yeniden yapılandırılmasında da yaygın olarak kullanılmaktadır.

3.3.3. Pekiştirmeli öğrenme

Bu öğrenme stili diğerlerinden biraz farklıdır. Canlıların davranış psikolojisine dayanır. Bu yöntem geri besleme ile öğrenme sürecini gerçekleştirir. Bu yöntem, olası durumların hedef olup olmadığını kontrol etmek için kullanılır. Denemeler sonucunda, hedefe ulaşılmadığında bir ödüsüz sinyal ve hedefe ulaşıldığında bir ödül sinyali alınır. Sistem bir ödüsüz sinyali aldığında hareketi tekrar etmez. Ödül sinyali aldığı deneyimlerden öğrenmeye devam eder ve daima maksimum ödül amacı ile sürekli öğrenme için çalışmaya devam eder. Bu nedenle, mükemmelliğe ulaşmak ve öğrenmeyi durdurmak gibi bir şey yoktur. Algoritma sürekli öğrenmeye devam etmektedir. Bu yöntemin temel amacı, insan beynine ve algılarına yakın (veya çok daha iyi bir seviyede) çalışan algoritmalar geliştirmektir. Bu nedenle, Derin Öğrenme (DL) algoritmaları pekiştirilmiş öğrenmenin mantığı ile geliştirilmiştir.

3.4. Optimizasyon ve Meta-sezgisel yaklaşımlar

Yöneylem araştırması, belirli bir problem için en uygun çözümü bulmayı içeren bir yöntem setidir. Dünyanın gerçek karmaşık problemlerinin çözümü için matematiksel modelleme, algoritmalar ve istatistikler gibi bilimsel yöntemleri kullanan bir bilim alanıdır. Probleme en uygun çözümü bulmak optimizasyondur. En uygun çözümü bulmak için gereken problemler Optimizasyon Problemleridir. Genellikle, NP-Complete karar problemleri gibi problemleri çözmek için kullanılır.

Optimizasyon problemleri, objektif fonksiyonumuzu minimum veya maksimum yapmaktır. Başka bir deyişle, bir şirketin optimizasyonunda verimliliği artırmak, maksimuma çıkarmaktır. Bir şirkette bir işin süresini azaltırken minimuma indirmektedir. Aslında, optimizasyon durumuyla günlük yaşamımızda birçok kez karşılaşılmaktadır.

Optimizasyon problemleri matematiksel modeldeki karar değişkenlerinin yapısına göre ayrık veya sürekli optimizasyona bölünmüştür. Bazı yerlerde, ayrık optimizasyonu birleşimsel olarak görebilirsiniz. Kesikli optimizasyon değişkenleri sınırlı bir set içerir. Örneğin, ünlü gezgin satıcı problemi ayrık optimizasyonla çözülmüştür. Sürekli optimizasyon problemleri kısıtlayıcı olmayan değişkenlerdir.

Sezgisel optimizasyon algoritmaları, optimizasyon problemlerine çözümler bulmak için operasyonel araştırmada kendisine bir yer bulmuştur. Çözülmesi gereken gerçek bir problemin önce matematiksel bir modele oturtulması gerekir. Bu nedenle, bu matematik formülü bir kriter olarak kullanarak, kısa sürede en iyi çözümü bulmayı amaçlamaktadır. Sezgisel optimizasyon algoritmaları mükemmel çözümü bulmayı garanti etmez.

1986 yılında, ilk olarak bahsedilen meta-heuristic terimi, Yunanca "meta" kelimesi ile "heuristic" kelimesinin bir kombinasyonudur ve "daha sezgisel" veya "daha yüksek sezgisel" olarak adlandırılır. Üst düzey sezgisel yaklaşım, çözüm alanındaki olasılıksal ancak bilinçli akıl yürütmeye dayanan yöntemleri içerir. Bu yöntemler, her adımda yaratılan çözüm kümesini temel alan yeni çözümler üretir. Bu şekilde, en uygun arama alanına en yakın noktalarda aramalar yapılır ve yerel en iyi nokta seçiminden kaçınılarak en iyi çözüm önerilir. Meta-sezgisel yöntemlerin karakteristik özellikleri aşağıdaki gibi ifade edilebilir (Blum & Roli, 2003):

1. Arama yöntemleri yol tarifleridir.

2. Amaç, en iyi veya en yakın sonuçları elde etmek için arama alanını etkin bir şekilde araştırmaktır.
3. Yerel arama tekniklerinden karmaşık öğrenme süreçlerine kadar çeşitlilik gösterir.
4. Genellikle belirleyici olmayan, yaklaşık bir çözüm sunarlar.
5. Arama alanındaki en iyi yerel konumlara takılmayı önleyen yapıları mevcuttur.
6. Belirli bir soruna özgü değildir.

Yöntemin temel kavramları probleme iyi adapte edilmelidir. Bu durumda meta-sezgisel algoritmalar iyi sonuçlar üretebilir (A. Hertz & Widmer, 2003). Meta-sezgisel algoritmalar genellikle beş sınıfa ayrılır (Tarantilis, Ioannou, & Prastacos, 2005):

- a) Doğa-ilham / ilham vermeyen algoritmalar
- b) Popülasyon tabanlı / Tek nokta (yerel arama) algoritmalar
- c) Dinamik/ Statik amaç fonksiyonlu algoritmalar
- d) Tek / Çok komşu algoritmalar
- e) Belleği olan veya olmayan algoritmalar

Biyoloji, fizik, zooloji ve bilgisayar bilimi gibi sosyal bilimlere dayalı çeşitli yöntemler geliştirmek son zamanlarda artan bir ivme kazanmıştır. Evrimsel algoritmalar mutasyon, çaprazlama ve doğal seleksiyon/seçim gibi özellikleri ile evrimin biyolojik teoreminden ilham almışlardır. Genetik algoritma, en çok bilinen evrimsel algoritmalarından biridir (Davis). Sürü zekası algoritmaları, sürülerin doğadaki hareketlerinden ilham alan arama algoritmalarıdır. Sürü üyeleri belirli bir sorunu çözmek için birbirleriyle etkileşime girerler. Parçacık sürü optimizasyonu (Eberhart & Kennedy, 1995), yapay arı kolonisi (Karaboga, 2005), karınca kolonisi (Dorigo & Gambardella, 1997), ateş böceği algoritması (Yang, 2010a), göçmen kuşlar optimizasyonu (Duman, Uysal, & Alkaya, 2012), yarasalar algoritması (Yang, 2010b), yapay alg algoritması (Uymaz, Tezel, & Yel, 2015), meyve sineği algoritması (Pan, 2012), yerçekimsel arama algoritması (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009) ve gri kurt optimizasyonu (Mirjalili, Mirjalili, & Lewis, 2014) gibi algoritmalar iyi bilinen sürü algoritmalarıdır.

3.5. Hibrit / Melez algoritmanın (GWO-NN) alt yapısı

Hibrit algoritmalar, araştırmanın genel verimliliğini artırmak veya iyileştirmek için iki veya daha fazla algoritmayı birleştirme işlemi ile üretilir. Bilim insanları genellikle bireysel algoritmaların faydalarını ortak yarar için kullanmaya çalışırlar. Uygulamada, bir hibritin performansı gerçekten geliştirip iyileştiremediğini ve farklı algoritmaları birleştirmenin yollarını bulabileceklerini bilmek zordur. Bu problem yeni algoritmalar uygulamada açık bir sorun olmaya devam etmektedir.

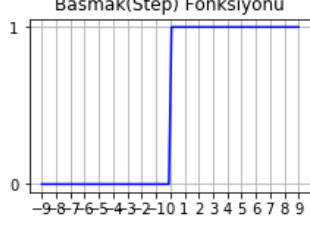
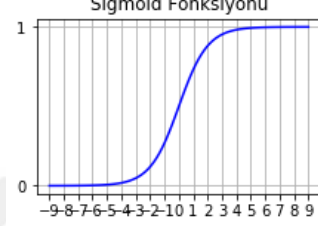
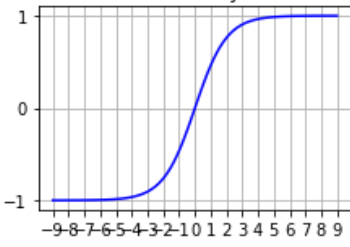
3.5.1. Sinir ağları (Neural networks)

Yapay Sinir Ağları (YSA), makine öğrenmesinde temel ve önemli sınıflandırma algoritmalarından biridir. YSA kavramı, 1943'te Warren McCulloch ve Walter Pitts tarafından önerilmiştir (McCulloch & Pitts, 1990). Bu algoritmanın ilham kaynağı, insan beyni nöronlarının nasıl çalıştığının (yani insan beyni hücrelerinin veya nöronların birbirine nasıl bağlandığı ve öğrendiği) modellenmesidir. Nöronların çeşitli şekilleri olmasına rağmen, elektrik sinyallerini dendritlerden aksonlara bir uçtan diğerine iletirler. Nöronlar, bir elektrik sinyalini alır ve başka bir elektrik sinyali yayarlar. Bu sinyaller daha sonra başka bir nörona iletilir.

Gözlemler nöronların hızlı tepki göstermediğini vurgulamakta, bunun yerine girdiyi belli bir eşiği geçene kadar baskı altına almakta ve bu da bir çıktıyı tetiklemektedir. Bu nedenle, herhangi bir çıktı oluşturulmadan önce ulaşılması gereken bir eşik vardır. Eşik değerini dikkate alarak çıktı üreten fonksiyonlara, Tablo 3.1'de gösterildiği gibi Aktivasyon Fonksiyonu adı verilir. Aktivasyon fonksiyonunun amacı, lineer olmayan çıkışa sahip bir nöron üretmektir, çünkü gerçek verilerin çoğu doğrusal değildir (non lineer). Nöronların bu doğrusal olmayan ilişkiyi verilerden öğrenmesi beklenir.

İnsan olarak örneklerle öğrenme yapmaktayız, sonra da benzer problemleri sınıflandırmaktayız ve bu işleyiş YSA için de geçerlidir. YSA, verilen veri setinin girişi ve çıkışı arasındaki ara bağlantıyı keşfetmeye çalışır. Farklı araştırmacılar farklı YSA versiyonları önermişlerdir. En popüler türleri; Kohonen self-organizing network (Kohonen, Oja, Simula, Visa, & Kangas, 1996), Radial basis function (RBF) network (Park & Sandberg, 1993), Spiking neural networks (Ghosh-Dastidar & Adeli, 2009), Feed-forward network (Bebis & Georgiopoulos, 1994) ve Recurrent neural network dır (Dorffner, 1996).

Tablo 3.1. İyi bilinen aktivasyon fonksiyonları

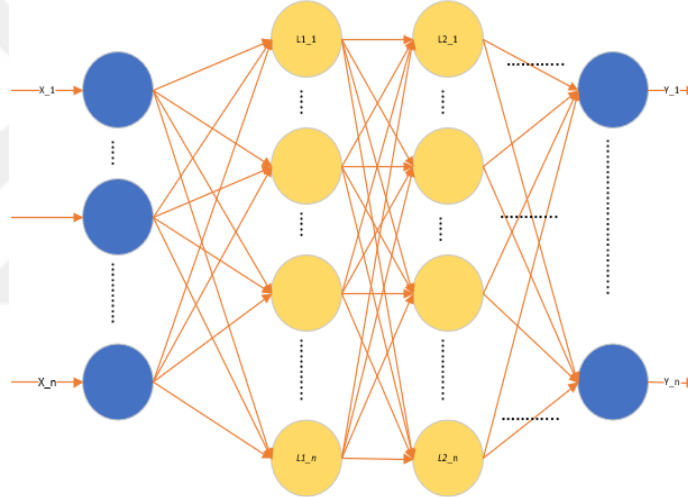
Fonksiyon	Denklemleri	Fonksiyonlar görselleri	Açıklama
Basamak (Step)	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$		Girdi 0'dan küçükse, daima 0'dır, aksi takdirde girdi 1'dir
Sigmoid	$f(x) = 1 / (1 + e^{-x})$		Genellikle ikili tahminler yapıldığında çıkış düğümünde aktivasyon fonksiyonu olarak kullanılır.
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$		Negatif değerleri sıfır ile değiştirir ve pozitif değerleri için x değerlerini tutar
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$		Gerçek değer girişini alır ve çıkış olarak [-1, 1] aralığına değiştirir

İleri beslemeli sinir ağı (FFNN) en basit türdür ve ilk geliştirilen NN'dir. Düğüm içeren katmanlar halinde düzenlenmiş çoklu nöronlar bulunmaktadır. Bu düğümlerin bir bağlantı veya bağlantısı mevcuttur. Her düğüm, diğer düğümler için nispeten önemli olan bir ağırlığa sahiptir. İleri beslemeli ağlarda, bilgiler yalnızca gizli düğümler tarafından ileri / giriş düğümleri ve çıktı düğümleri arasında ileri yönde taşınır. Yani, ağda hiçbir döngü yoktur. İleri besleme üç katmandan oluşur; giriş, gizli ve çıkış katmanları. Bu katmalar aşağıdaki gibi açıklanabilir:

- Giriş düğümleri ağa bilgi sağlar ve Giriş Katmanları (**Input Layers**) olarak adlandırılır. Bu düğümlerde hesaplama yapılmaz. Bilgileri Gizli düğüme iletirler.

- b) Gizli düğümlerin dış dünyayla doğrudan bağlantısı yoktur. Girişten çıkış düğümlerine bilgi / veri hesaplar ve iletir. Bu düğümler Gizli Katmanlar (**Hidden Layers**) olarak bilinir. İleri beslemede, yalnızca bir giriş katmanı ve bir çıkış katmanı bulunurken, sıfır veya daha fazla gizli katman bulunabilir.
- c) Çıkış düğümleri toplu olarak Çıkış Katmanı (**Output Layer**) olarak adlandırılır. Hesaplamalardan ve bilgileri ağdan dışarıya aktarmaktan sorumludurlar.

İleri besleme iki türden oluşur; gizli katmanı olmayan tek bir algılayıcı ve bir veya daha fazla gizli katmanı olan Çok Katmanlı Algılayıcı (MLP). Şekil 3.7, 2 gizli katmanı olan NN'yi göstermektedir. BackProp olarak kısaltılmış hataların geri yayılması algoritması, yapay sinir ağını eğitmek için çeşitli yollardan biridir. Basitçe, hatalardan öğrenir ve düğümler arasındaki ağırlıkları optimize eder.



Şekil 3.7. Çok katmanlı algılayıcı için yapay sinir ağı.

Bu tez çalışmasında da, çok katmanlı algılayıcı ileri beslemeli sinir ağı (MLP) uygulanmıştır. MLP sinir ağına kullanılan denklem aşağıda verilmiştir (Mirjalili, 2015):

A. İlk önce girdiler ağırlıklı toplamalar şeklinde Denklem 3.1 ile hesaplanır

$$S_j = \sum_{i=1}^n W_{ij} \cdot X_i - \theta_j, \quad j = 1, 2, 3, \dots, h \quad (3.1)$$

burada n , girişlerin sayısıdır, W_{ij} , giriş katmanındaki i 'den gizli katmandaki j 'e olan bağlantı ağırlığını gösterir, X_i , i girişi olarak ifade eder ve θ_j , j gizli düğümün bias'dır.

B. Her gizli nöronun çıktısı Denklem 3.2 kullanılarak hesaplanır:

$$S_j = \text{sigmod}(S_j) = \frac{1}{(1 + \exp(-S_j))}, j = 1, 2, 3, \dots, h \quad (3.2)$$

C. Çıktı(lar) Denklem 3.3 ve 3.4 kullanılarak gizli nöronların çıktısına dayanarak hesaplanır:

$$O_k = \sum_{j=1}^h W_{jk} \cdot S_j - \theta_k, \quad k = 1, 2, 3, \dots, m \quad (3.3)$$

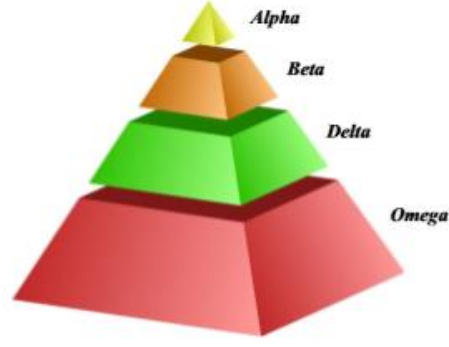
$$O_k = \text{sigmod}(O_k) = \frac{1}{(1 + \exp(-O_k))}, \quad k = 1, 2, 3, \dots, m \quad (3.4)$$

Burada W_{jk} j gizli katmanından k çıktı katmanına bağlantı ağırlığı ve θ_k , k çıktı katmanının bias'dır.

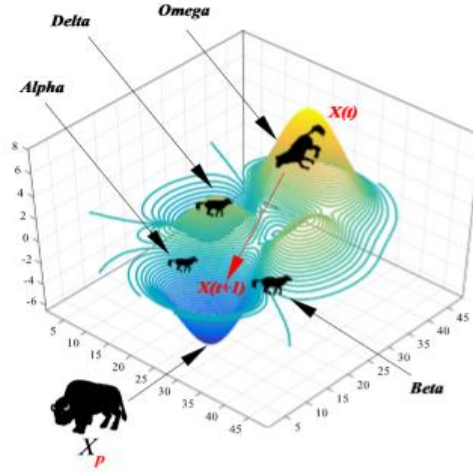
3.5.2. Gri kurt optimizasyonu (Grey wolf optimization GWO)

GWO, Mirjalili tarafından 2014 yılında önerilen en yeni ve hızla kullanımı genişleyen sürü zekâsı algoritmalarından biri olarak kabul edilmektedir (Mirjalili et al., 2014). GWO, küresel optimizasyon problemleri, elektrik ve enerji mühendisliği problemleri, zamanlama problemleri, elektrik dağıtım problemleri, kontrol mühendisliği problemleri, robotik veya yol planlama problemleri, çevre planlama problemleri ve diğerleri gibi farklı problemleri çözmek için kullanılmıştır. GWO algoritmasının ilham kaynağı, gri kurtların liderlik ve avlanmadaki sosyal zekâsıdır. Gri kurtların popülasyonu dört grup içerir: alfa (α), beta (β), delta (δ) ve omega (ω). Her gri kurt sürüsünde, gücü ve egemenliği belirleyen ortak bir sosyal hiyerarşi vardır (bkz. Şekil 3.8).

En güçlü kurt; avcılık, göç, beslenme, uyuma yeri, uyanma zamanı ve vb tüm sürüyü yönlendiren alfadır. Alfa kurdu liderlik etmek için çok yaşlandığında, hasta ya da ölü olduğunda, Alfa kurt un yerini alabilecek en büyük ihtimalli kurt Beta Kurttur. Delta ve omega'nın gücü ve egemenliği, Şekil 3.8 ve Şekil 3.9'de görülebileceği gibi alfa ve beta'dan daha azdır. Delta kurtlara, ast denir ve nöbetçilerden, avcılardan, yaşlılardan, bekçilerden ve keşiflerden/izciler oluşur. İziciler bölgenin sınırlarını gözetir ve sürüyü herhangi bir riske karşı uyarırlar. Omega kurtları günah keçisi olarak kabul edilmektedir. Alfa, beta ve delta, omega kurtlarını arama alanının gelecek vaat eden bölgelerine yönlendiren en iyi arama aracı olarak kabul etmektedir (Elhariri, El-Bendary, Hassanien, & Abraham, 2015). Gri kurtlar avlanırken, bir dizi etkili adımı takip ederler: kovalama, kuşatma, çok yorma / arama ve saldırı. Bu adımlar çok iyi avlamalarını sağlamaktadır. GWO'nun matematik modeli Şekil 3.8 ve 3.9 de tanımlanmıştır:



Şekil 3.8. Gri kurt hiyerarşisi



Şekil 3.9. Alfa, beta, delta ve omeganın grafiksel tanımı

3.5.2.1. Etrafını çevirme

Denklemler 3.5 ve 3.6 (Mirjalili, 2015) avın etrafını sarma formülü ve yapılandırılması için tanımlanmıştır.

$$D = |C \cdot X_p(t) - X(t)| \quad (3.5)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (3.6)$$

Burada $X(t)$ kurdun konumu ve t döngü sayısıdır. X_p av pozisyonu ve D Denklem 1 kullanılarak hesaplanmaktadır. A ve C , sırasıyla Denklemler 3.7 ve 3.8 'e göre hesaplanan katsayılardır.

$$A = 2a \cdot r_1 - a \quad (3.7)$$

$$C = 2r_2 \quad (3.8)$$

Burada r_1 ve r_2 , optimum çözümü ortaya çıkarmak için kullanılan 0 ile 1 arasındaki rastgele vektörlerdir.

3.5.2.2. Av aşaması

Avlanma, aşağıdaki denklemde belirtildiği gibi av konumunu bilmek için gerekli olan α , β ve δ 'dan gelen bilgiler kullanılarak gerçekleştirilir.

$$X(t + 1) = \frac{X_1 + X_2 + X_3}{3} \quad (3.9)$$

Burada X_1 , X_2 ve X_3 değerleri sırasıyla Denklem 3.10, 3.11 ve 3.12 kullanılarak hesaplanmaktadır.

$$X_1 = X_\alpha - A_1 \cdot (D_\alpha) \quad (3.10)$$

$$X_2 = X_\beta - A_2 \cdot (D_\beta) \quad (3.11)$$

$$X_3 = X_\delta - A_3 \cdot (D_\delta) \quad (3.12)$$

Döngünün t anında, popülasyondaki en iyi üç çözüm sırasıyla X_1 , X_2 ve X_3 'tür. A ve C denklemleri yukarıda belirtilmiştir. D_α , D_β ve D_δ buna göre Denklem 3.13, Denklem 3.14 ve Denklem 3.15'te hesaplanır. Diğer tüm kurtlar konumlarını α , β ve δ konumlarını kullanarak adapte ederler.

$$D_\alpha = |C_1 \cdot X_\alpha - X| \quad (3.13)$$

$$D_\beta = |C_2 \cdot X_\beta - X| \quad (3.14)$$

$$D_\delta = |C_3 \cdot X_\delta - X| \quad (3.15)$$

3.5.2.3. Ava saldırma aşaması

Doğrusal olarak azalan bir parametre olan a , döngü sayısı boyunca 2 ila 0 arasındadır. a 'nın formülü Denklem 3.16 da verilmiştir.

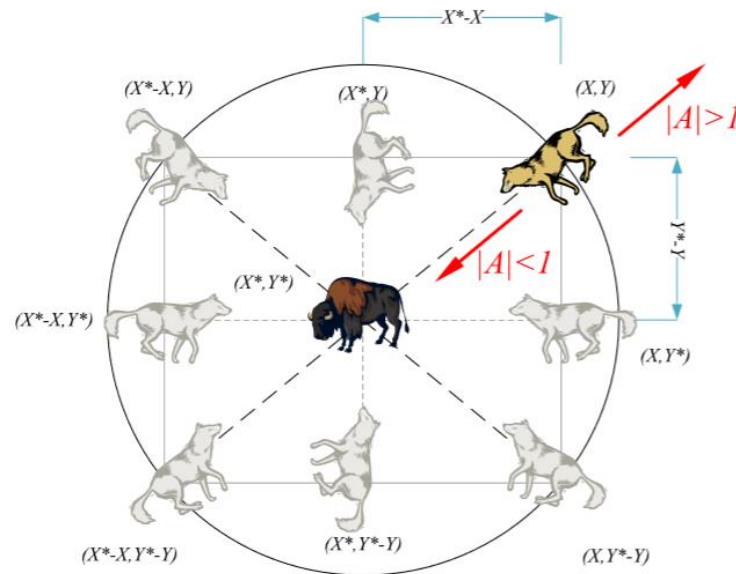
$$a = 2 - t(2/NLoops) \quad (3.16)$$

Burada t döngü numarası ve $NLoops$, döngülerin toplam sayısıdır. a parametresi, sürüde exploitation(sömürü) ve exploration(keşif) arasındaki dengeyi kontrol etmek için kullanılır. Kurtların yarısı keşif, diğer yarısı sömürü için kullanılır.

Bu nedenle, GWO, sömürü ve keşif arasında sorunsuzca geçiş yapabilir. Denklem 3.7'de A , $[-a, a]$ aralığında rastgele bir değerdir. $[a]$ azaldığında, A da azalır. A $[-1, 1]$ olduğunda, bir sonraki arama kümesinin yeri, mevcut konum ile av konumu arasındaki herhangi bir konum olabilir (yani, $A < 1$ ve $C < 1$ olduğunda sömürü gerçekleşir).

3.5.2.4. Arama aşaması

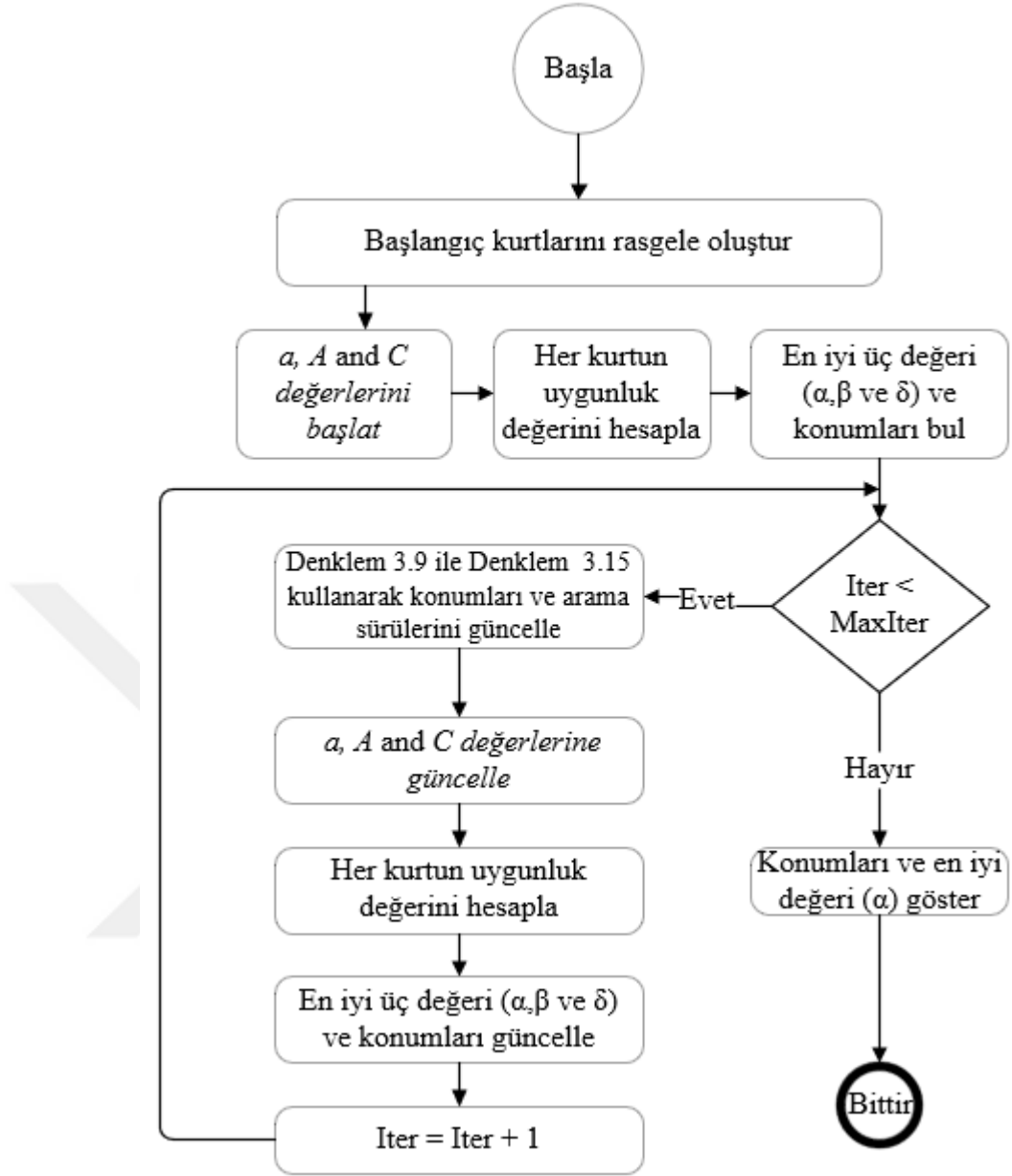
GWO'da optimum arama α , β ve δ konumlarına bağlıdır. Kurtlar av ararken birbirlerinden uzaklaşırlar ve daha iyi bir av keşfettiklerinde tekrar saldırmak için birleşirler. Ayrıca, A 'nın rastgele değeri -1 'den küçük veya 1 'den büyük olduğunda ($A < -1$ veya $A > 1$), arama sürüsü, Şekil 3.10'da gösterildiği gibi avlamaya ayrılır. Başka bir deyişle, keşif ($A < -1$ veya $A > 1$) olduğunda gerçekleşir. GWO'nun bir başka önemli parametresi, Denklem 3.8'deki 0 ila 2 arasında olan C 'dir ($C > 1$) 'de keşif işlemi sırasında kurtlara yardımcı olmaktadır. C , Denklem 3.5'deki mesafeyi etkiler. Yani, bu GWO'nun arama ve yerel optimuma takılmama için daha rastgele davranışlar ürettiğini ifade etmektedir.



Şekil 3.10. Sürülerinin arama konum güncelleme tekniği ve A 'nın üzerindeki etkileri.

GWO algoritmasının çalışma modeli aşağıdaki gibidir (Mirjalili et al., 2014):

- 1) Üst ve alt sınıra bağlı olarak rastgele bir kurt sürüsü başlat
- 2) Her kurt için karşılık gelen nesnel değeri hesapla
- 3) İlk en iyi 3 kurdu seç ve α , β ve δ olarak sakla.
- 4) Denklem 3.9 – Denklem 3.15 denklemlerini kullanarak sürünün geri kalanının konumunu (ω) güncelle.
- 5) a , A ve C parametrelerini güncelle
- 6) Son kriter elde edilemezse veya iterasyon sonlanmadı ise, 2. adıma geri dön.
- 7) α konumunu, tahmin edilen en iyi optimum olarak döndür.



Şekil 3.11. GWO'nun akış şeması

3.6. Kullanılan Veri Setleri

Makine öğrenmesi yöntemleri örneklerle öğrenmeye odaklıdır. Bu nedenle veriyi tanımlamak için kullanılan girdi verilerini ve terminolojiyi tanımak önemlidir. Veri kümesi aslında bir dizi örnektir ve makine öğrenmesi algoritmalarını kullanırken, genellikle farklı amaçlar için çeşitli verilere ihtiyaç duyulur. Bu tez çalışmasının amacı güvenlik olduğu için kullanılan veri kümesi de güvenlik amaçlıdır. Bu yüzden, tezde NSL-KDD ve UNSW-NB15 gibi en son saldırı tespit veri setleri kullanılmıştır.

2009'da NSL-KDD adlı yeni bir KDD99 sürümü yayınlanmıştır (Tavallaee et al., 2009). KDD99'un boyutu azaltılmış ve tekrarlanan veriler kaldırılmıştır. Veri setinde eğitim için 125973 ve test için 22544 örneklem mevcuttur. Bu veri setinin farklı konfigürasyonları, olay sayısındaki varyasyonla elde edilmiştir, ancak her durumda özniteliklerin sayısı 42 özniteliktir. Veri setinin 42.ci özniteliği, belirli bir olayın normal bir bağlantı olayı mı yoksa saldırı mı olduğunu gösteren "sınıf" özniteliğidir. Sınıf niteliği beş sınıfa ayrılır; bir normal ve dört saldırıdır (Probe, DoS, R2L ve U2R). İlk 41 nitelik, Tablo 3.2'de gösterildiği gibi dört farklı sınıfa ayrılabilir.

- Temel Özellikler (T), bireysel TCP bağlantılarının özellikleridir.
- İçerik özellikleri (İ), alan bilgisi tarafından önerilen bir bağlantı içindeki özelliklerdir.
- Trafik özellikleri (TT), iki saniyelik bir zaman penceresi kullanılarak hesaplanan özelliklerdir.
- Host özellikleri (H), iki saniyeden uzun süren saldırıları değerlendirmek için tasarlanmış özelliklerdir.

Tablo 3.2. NSL-KDD Veri seti özellikleri

No	Özellik	Açıklama	Sınıf	Çeşit
1	Duration	Bağlantı süresi	T	Sürekli
2	Protocol_type	Protokol türü, örneği. TCP, UDP, vb.	T	Ayrık
3	Service	Hedef ağ hizmeti, örneği. HTTP, telnet vb.	T	Ayrık
4	Src_bytes	Kaynaktan hedefe aktarılan veri miktarı (bayt).	T	Sürekli
5	Dst_bytes	Kaynağa iletilen veri miktarı (bayt)	T	Sürekli
6	Flag	Bağlantının durumu: normal veya hata.	T	Ayrık
7	Land	Bağlantının kaynağı ve hedefi aynı sunucu / port ise 1, aksi takdirde 0.	T	Ayrık
8	Wrong_fragment	Arızalı parça sayısı	T	Sürekli
9	Urgent	Acil durum paket sayısı	T	Sürekli
10	Hot	"Sıcak" gösterge sayısı	İ	Sürekli
11	Num_failed_logins	Başarısız oturum açma sayısı	İ	Sürekli
12	Logged_in	Başarılı giriş ise 1; aksi takdirde 0	İ	Sürekli
13	Num_compromised	"Tehlikeli" durum sayısı	İ	Sürekli
14	Root_shell	Root shell elde edildiyse 1; aksi takdirde 0	İ	Ayrık
15	Su_attempted	"Su root " komutu çalıştırıldıysa 1; aksi takdirde 0	İ	Ayrık
16	Num_root	Root'un erişim sayısı	İ	Sürekli
17	Num_file_creations	Oluşturulan dosya sayısı	İ	Sürekli

18	Num_shells	Shell'in açık sayısı	İ	Sürekli
19	Num_access_files	Erişim kontrol dosyalarındaki işlem sayısı	İ	Sürekli
20	Num_outbound_cmds	FTP oturumundaki giden komutların sayısı	İ	Sürekli
21	Is_hot_login	"Sıcak" listeden ilk girişi ise 1; aksi takdirde 0	İ	Ayrık
22	Is_guest_login	"Misafir" girişi ise 1; aksi halde 0	İ	Ayrık
23	Count	t sn içinde aynı sunucuya yapılan bağlantı sayısı	TT	Sürekli
24	Serror_rate	"SYN" hatasıyla alınan bağlantıların yüzdesi	TT	Sürekli
25	Rerror_rate	"REJ" hatasıyla alınan bağlantıların yüzdesi	TT	Sürekli
26	Same_srv_rate	Aynı hizmete yapılan bağlantıların yüzdesi	TT	Sürekli
27	Diff_srv_rate	Farklı hizmetlere yapılan bağlantıların yüzdesi	TT	Sürekli
28	Srv_count	t saniye içinde aynı hizmete yapılan bağlantı sayısı	TT	Sürekli
29	Srv_serror_rate	"SYN" hatasıyla alınan bağlantıların yüzdesi	TT	Sürekli
30	Srv_rerror_rate	"REJ" hatasıyla alınan bağlantıların yüzdesi	TT	Sürekli
31	Srv_diff_host_rate	Farklı sunuculara yapılan bağlantıların yüzdesi	TT	Sürekli
32	Dst_host_count	Aynı sunucuya yapılan n bağlantıda yapılan bağlantı sayısı	H	Sürekli
33	Dst_host_srv_count	Aynı sunucuya yapılan n bağlantıda yapılan bağlantı sayısı	H	Sürekli
34	Dst_host_same_srv_rate	Aynı hizmete yapılan bağlantıların yüzdesi	H	Sürekli
35	Dst_host_diff_srv_rate	Farklı hizmetlere yapılan bağlantıların yüzdesi	H	Sürekli
36	Dst_host_same_src_port_rate	Aynı kaynak port bağlantısından gelen bağlantıların yüzdesi.	H	Sürekli
37	Dst_host_srv_diff_host_rate	Farklı sunuculara yapılan bağlantıların yüzdesi	H	Sürekli
38	Dst_host_serror_rate	"SYN" hatasıyla alınan bağlantıların yüzdesi	H	Sürekli
39	Dst_host_srv_serror_rate	"SYN" hatasıyla alınan bağlantıların yüzdesi	H	Sürekli
40	Dst_host_rerror_rate	"REJ" hatasıyla alınan bağlantıların yüzdesi	H	Sürekli
41	Dst_host_srv_rerror_rate	"REJ" hatasıyla alınan bağlantıların yüzdesi	H	Sürekli

Nour ve Slay, 2015 yılında IXIA PerfectStorm araçlarını kullanarak UNSW-NB15 adlı saldırı tespiti için yeni bir veri seti oluşturmuşlardır (Moustafa & Slay, 2015). Bu veri seti 49 nitelik içermektedir. Son sütun dokuz saldırı (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms) ve 1 normal içeren sınıf değeridir. Veri setinde eğitim için 175341 ve test için 82332 satır bulunmaktadır. Sınıf özniteliğiyle toplam 49 özellik oluşturmak için Bro-IDS araçlarını

kullanmışlar ve on iki algoritma geliştirmişlerdir. UNSW-NB15 özniteliklerinin ayrıntıları Tablo 3.3'te açıklanmıştır.

Tablo 3.3. UNSW-NB15 Veri seti özellikleri

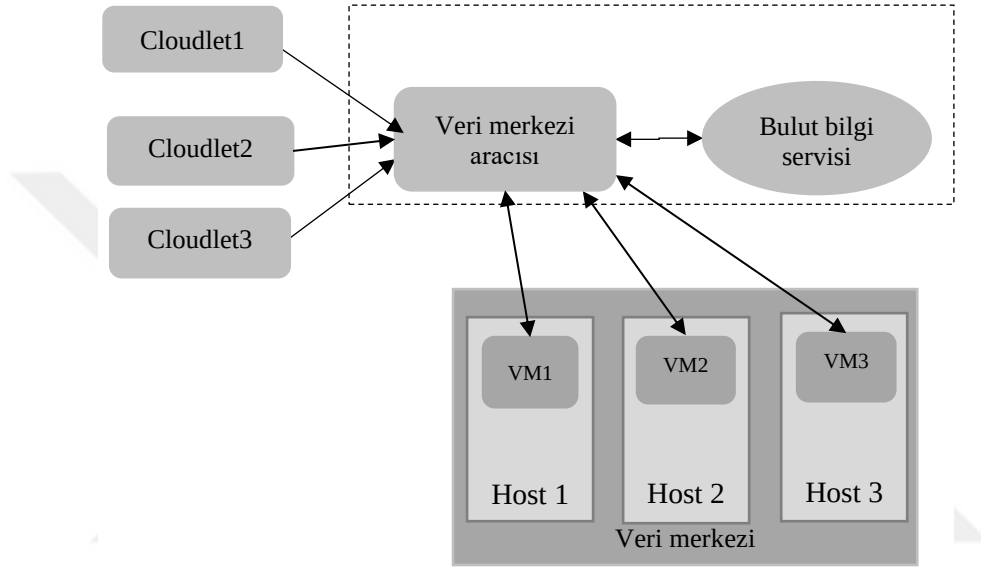
No	Özellik	Açıklama	Tip
1	srcip	Kaynak IP adresi	Nominal
2	sport	Kaynak port numarası	Integer
3	dstip	Hedef IP adresi	Nominal
4	dsport	Hedef port numarası	Integer
5	proto	İşlem protokolü	Nominal
6	state	Durumu ve bağımlı protokolünü gösterir, örneğin. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN ve (-) (durumu kullanılmıyorsa)	Nominal
7	dur	Kayıt toplam süresi	Float
8	sbytes	Kaynaktan hedefe işlem baytları	Integer
9	dbytes	Hedekten kaynağa işlem baytları	Integer
10	sttl	Kaynaktan hedefe değer yaşama süresi	Integer
11	dttl	Hedekten kaynağa değer yaşama süresi	Integer
12	sloss	Kaynak paketler yeniden iletili veya düşürüldü	Integer
13	dloss	Hedef paketleri yeniden iletili veya düşürüldü	Integer
14	service	http, ftp, smtp, ssh, dns, ftp-data, irc ve (-) çok kullanılan servis değilse	Nominal
15	Sload	Saniyedeki kaynak bit sayısı	Float
16	Dload	Saniyedeki hedef bit sayısı	Float
17	Spkts	Kaynaktan hedefe paket sayısı	Integer
18	Dpkts	Hedekten kaynağa paket sayısı	Integer
19	swin	Kaynak TCP penceresi reklam değeri	Integer
20	dwin	Hedef TCP penceresi reklam değeri	Integer
21	stcpb	Kaynak TCP temel sıra numarası	Integer
22	dtcpb	Hedef TCP temel sıra numarası	Integer
23	smeansz	Src tarafından iletilen akış paket boyutunun ortalaması	Integer
24	dmeansz	Dst tarafından iletilen akış paket boyutunun ortalaması	Integer
25	trans_depth	http talebi / yanıt işleminin bağlantısının derinliği	Integer
26	res_bdy_len	Sunucunun http hizmetinden aktarılan verilerin gerçek sıkıştırılmamış içerik boyutu.	Integer
27	Sjit	Kaynak titreşimi (mSn)	Float
28	Djit	Hedef titreşimi (mSn).	Float
29	Stime	kayıt başlangıç zamanı	Timestamp

30	Ltime	kayıt bitiş zamanı	Timestamp
31	Sintpkt	Kaynak paketler arası varış zamanı (mSn)	Float
32	Dintpkt	Hedef paketler arası varış zamanı (mSn)	Float
33	tcprtt	TCP'nin 'synack' ve 'ackdat' toplamı	Float
34	synack	TCP'nin SYN ve SYN_ACK paketleri arasındaki süre.	Float
35	ackdat	TCP'nin SYN_ACK ve ACK paketleri arasındaki süre.	Float
36	is_sm_ips_ports	Kaynak (1), hedef (3) IP adreslerine ve port numaraları (2) (4) eşitse, bu değişken 1, yoksa 0 değerini alır	Binary
37	ct_state_ttl	Kaynak / hedef yaşam süresi (10) (11) için belirli değer aralığına göre her durum (6) için sayısı	Integer
38	ct_flw_http_mthd	Http hizmetinde Al ve Gönder gibi yöntemlere sahip akış sayısı.	Integer
39	is_ftp_login	Ftp oturumuna kullanıcı ve parola ile erişiliyorsa, 1'dir, aksi takdirde 0'dir	Binary
40	ct_ftp_cmd	Ftp oturumunda komut içeren akış sayısı.	Integer
41	ct_srv_src	Bitiş zamana (26) göre 100 bağlantıda aynı hizmeti (14) ve kaynak adresini (1) içeren bağlantı sayısı	Integer
42	ct_srv_dst	Bitiş zamana (26) göre 100 bağlantıda aynı hizmeti (14) ve hedef adresini (3) içeren bağlantı sayısı.	Integer
43	ct_dst_ltm	Bitiş zamana (26) göre 100 bağlantıda aynı hedef adresin (3) bağlantı sayısı.	Integer
44	ct_src_ltm	Bitiş zamana (26) göre 100 bağlantıda aynı kaynak adresinin (1) bağlantı sayısı.	Integer
45	ct_src_dport_ltm	Bitiş zamana (26) göre 100 bağlantıda aynı kaynak adresinin (1) ve hedef portun (4) bağlantı sayısı	Integer
46	ct_dst_sport_ltm	Son zamana (26) göre 100 bağlantıda aynı hedef adrese (3) ve kaynak porta (2) ait bağlantı sayısı	Integer
47	ct_dst_src_ltm	Son zamana (26) göre 100 bağlantıda aynı kaynak (1) ve hedef (3) adresinin bağlantı sayısı	Integer
48	attack_cat	Her saldırı kategorisinin adı. Bu veri setinde dokuz kategori mevcut (örneği. Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms)	Nominal
49	Label	0 normal için ve 1 saldırı kayıtları için	Binary

3.7. CloudSim

Bulut bilişim, güvenli, sürdürülebilir, hataya dayanıklı, ölçeklenebilir ve güvenilir IT hizmetleri sunmada öncü bir yaklaşımdır. Bu nedenle, yeni bulut uygulamalarının performansını ve stratejileri geliştirilmeden önce değerlendirmek için hızlı, tekrarlanabilir ve kontrol edilebilir metodolojilere ihtiyaç vardır. Gerçek test tezgahlarının kullanımı test ölçeği deneylerini sınırladığından ve sonuçların çoğaltılmasını oldukça zorlaştırdığından, simülasyon kullanılabilir.

CloudSim, Melbourne Üniversitesi Bilgisayar Bilimleri Bölümü tarafından geliştirilmiştir. CloudSim'in temel amacı, bulut bilişim altyapılarının ve ortaya çıkan uygulama hizmetlerinin denenmesi, modellenmesi ve simülasyonu için genelleştirilmiş ve genişletilebilir bir simülasyon platformu oluşturmaktır. Kullanıcıların, altyapıdaki ve bulut hizmetlerindeki zayıf detaylardan endişe etmeden çalışmak istedikleri belirli sistem tasarımlarına odaklanmalarını sağlamaktadır. Şekil 3.12, CloudSim'in genel mimarisini göstermektedir.



Şekil 3.12. CloudSim mimarisi

CloudSim'in kendisi bir grafik kullanıcı arayüzüne sahip değildir; CloudReports, CloudSimEx, EdgeCloudSim, WorkflowSim, CloudAnalyst, iFogSim ve Cloudsim Plus gibi uzantılar, CloudSim simülasyonları için bir arayüz sunmaktadır. Şekil 3.13'te, CloudReports arayüzü gösterilmektedir.



Şekil 3.13. CloudReports arayüzü

CloudSim'in detayları aşağıda sunulmuştur.

3.7.1. Veri merkezi

Veri merkezi bir dizi ana bilgisayardan oluşur ve sanal makineleri (VM) (örneğin sanal makineleri sağlama gibi) yönetmekten sorumludur. Broker VM istekleri alarak ve hostlarda VM'ler oluşturarak bir IaaS sağlayıcısı gibi davranır.

3.7.2. Veri merkez brokeri

Bu sınıf, kullanıcı adına hareket eden bir aracıyı temsil eder. İki mekanizmayı değiştirir: VM sağlama isteklerini veri merkezlerine gönderme ve bir diğerini de VM'lere görev gönderme mekanizması. CloudSim kullanıcıları, kendi kurallarını denemek için bu sınıfı genişletmelidir.

3.7.3. Host

Ana bilgisayar sanal makine yönetimi ile ilgili eylemleri gerçekleştirir (örneğin, oluşturma ve imha) ve sanal makinelerde görev işlemeyi günceller. Bir ana bilgisayarın sanal makineler için öğeleri işleme, bellek sağlama ve bant genişliği için belirli bir

politikası vardır. Bir ana bilgisayar bir veri merkezine bağlanır ve sanal makineleri barındırabilir.

3.7.4. Sanal makine

Fiziksel makineler olarak çalışan sanal makineler (VM) denilen uygulamaları çalıştıran bir makinenin yazılım uygulamasını temsil eder. Her sanal makine, ana bilgisayardan alınan kaynakları orada yürütülen görevler arasında böler.

3.7.5. Cloudlet

Bir cloudlet sınıfı da bir görev olarak adlandırılır. CloudSim, bir uygulamanın yapılandırma gereksinimleri açısından karmaşıklığına işaret eder. Bu sınıf, Veri merkezi broker sınıfında uygulanan planlama politikası tarafından yönetilir.

4. ÖNERİLEN GWO VE NN HİBRİT ALGORİTMASI

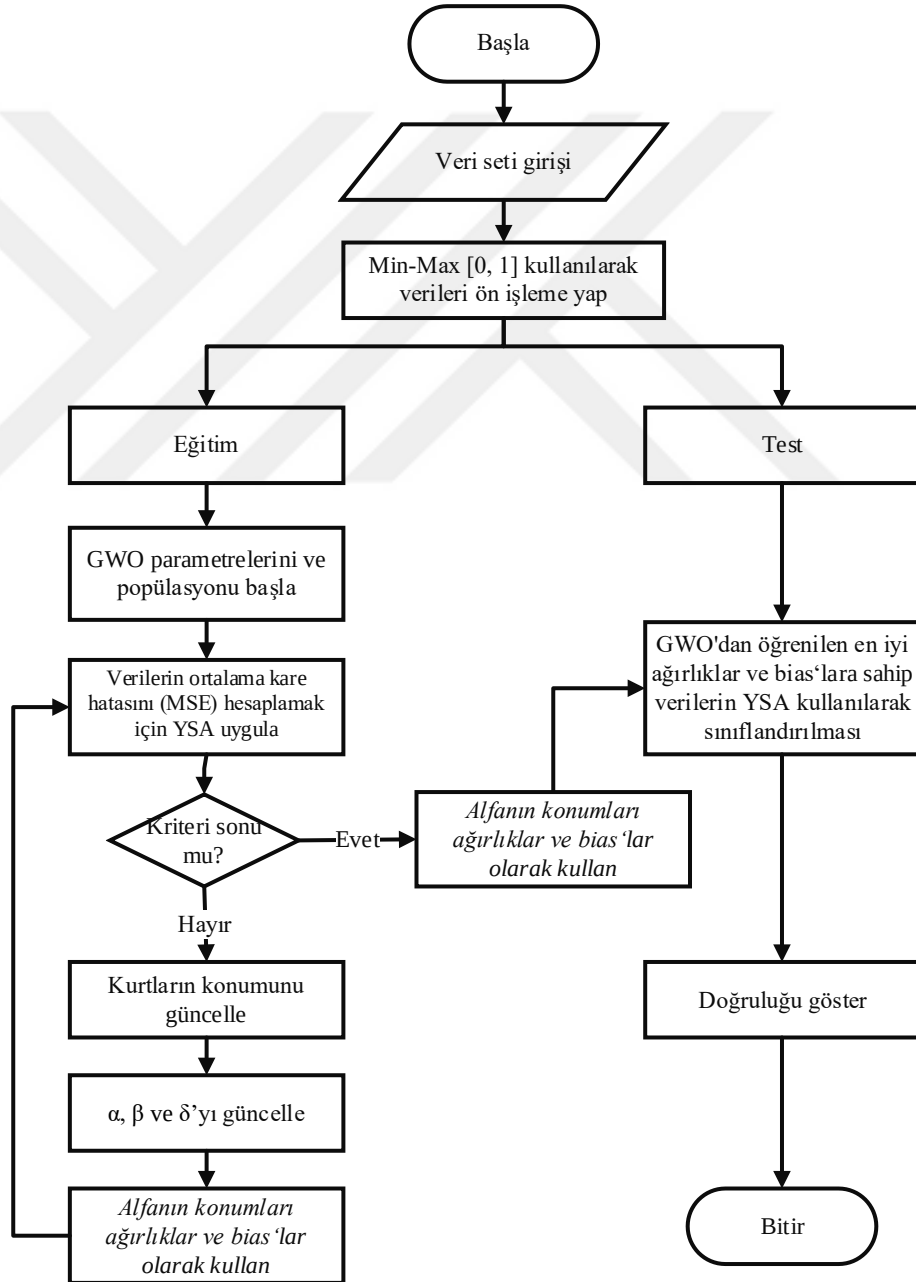
Yapay Sinir Ağları (YSA), makine öğrenimindeki en temel ve en önemli algoritmalarından biridir. YSA, yeni verilen girdi veri kümeleri üzerinde en yüksek performansı elde etmek için sinir ağlarını eğitmekten sorumlu bir eğitmen gibidir. Gözetimli öğrenmede bir eğitim yöntemi, önce sinir ağlarına eğitim örnekleri vermektedir. Eğitmen daha sonra performansı iyileştirmek için her eğitim aşamasında YSA'nın yapısal parametrelerini değiştirmektedir. Eğitmen, herhangi bir YSA'nın en önemli bileşeni olarak düşünülebilmektedir (Mirjalili, 2015).

Literatürde, deterministik ve stokastik olarak iki tür öğrenme yöntemi mevcuttur. Geri Yayılma (BP) ve gradyan tabanlı yöntemler deterministik yöntemler olarak varsayılmaktadır. Bu tür tekniklerde, eğitim örnekleri tutarlı kalırsa eğitim aşaması aynı performansla sonuçlanmaktadır (J. A. Hertz, Krogh, & Palmer). Bu gruptaki eğitmenler çoğunlukla minimum hata (maksimum performansı) bulmayı amaçlayan matematiksel optimizasyon metotlarıdır. Deterministik eğitmenlerin avantajları basitlik ve hızıdır. Deterministik eğitim algoritması genellikle bir çözümle başlar ve onu bir optimuma doğru yönlendirmektedir. Yakınsama çok hızlıdır, ancak elde edilen çözümün kalitesi büyük ölçüde ilk çözüme bağlıdır. Ek olarak, yerel optimuma takılma olasılığı yüksektir. Yerel optimum, yanlışlıkla global optimum olarak kabul edilebilecek bir arama alanındaki optimum altı çözümleri ifade etmektedir. Buradaki zorluk, bir eğiticinin global optimumu bulmak için farklı başlangıç çözümleriyle yeniden başlatılması gereken bilinmeyen uygulama sayısıdır.

Bunun aksine, stokastik eğitmenler eğitim sürecini rastgele bir çözüm(ler) ile başlatıp geliştirmektedir. Rastgelelik, eğitim sürecinde hem başlangıç çözümlerine hemde çözümün iyileştirme yöntemlerine uygulanan stokastik eğitmenlerin temel bileşenidir. Bu tür yöntemlerin avantajı, yüksek yerel optimumdan kaçınmadır, ancak bunlar çoğunlukla deterministik yaklaşımlardan çok daha yavaştır (Wang, Gandomi, Alavi, & Hao, 2014; Wang, Guo, Gandomi, Hao, & Wang, 2014). Literatürde, yüksek yerel optimumdan kaçınma amacıyla son zamanlarda stokastik eğitim yöntemlerinin çok ilgi çektiğini göstermektedir. Çoklu çözümlere sahip stokastik algoritmaların, yerel optimum durgunluğunu çözmek için daha yüksek yetenek sağladığı görülmektedir. Stokastik eğitmenler, bir NN'nin performansını en üst düzeye çıkarmak için stokastik optimizasyon metotlarını kullanmaktadır.

Birçok YSA türü vardır ve çok katmanlı algılayıcı (Multi-Layer Perceptron, MLP), bu sinir ağlarının bir çeşididir. YSA, saldırı tespitinde sıklıkla kullanılıp uygulanmaktadır. Bu alanda çok başarılı sonuçlar vermektedir. Çok katmanlı yapay sinir ağının (MLP) eğitimi, bilinmeyen arama alanı nedeniyle zorlu bir konudur. Bununla birlikte, eğitim süresinin uzunluğu pratikte sorunlara neden olabilmektedir. Ayrıca MLP'ye verilen veri kümelerini ve girdileri değişebilmektedir.

Tez çalışmasında, Saldırı tespiti çalışılan veri setlerinde Gri kurt ve Sinir ağlarının (GWO-NN) bir kombinasyonunun ne kadar etkili olduğu araştırılmıştır.



Şekil 4.1. GWO-NN hibrit yaklaşım mimarisi.

MLP eğitiminde ağırlıklar ve bias'lar en önemli değişkenlerdir. Bir eğitmen, en yüksek sınıflandırma doğruluğunu ve minimum hatayı sunan ağırlıklar ve bias'lar için bir yığın değer elde etmelidir. Geri yayılma algoritması, YSA'yı veri girişi ve çıkışı arasındaki ara bağlantıyı bulmak üzere eğitirken ağırlıkları ve bias'ları ayarlamak için kullanılan ilk yöntemlerden biridir. Bu nedenle, GWO en uygun ağırlık ve bias'ları elde etmek için kullanılmıştır. Ağırlıklar ve bias'lar bu algoritma için değişkenlerin bir vektörü olarak hesaplanmaktadır. Her vektörün uygunluğu, gerçek giriş ile istenen çıkış arasındaki hatayı hesaplayan Hataların karelerinin ortalaması (Mean Squared Errors, MSE) kullanılarak hesaplanmıştır. Düşük MSE, iyi model ve daha iyi doğruluk göstermektedir. MSE ortalaması denklem 4.1 kullanılarak hesaplanır:

$$MSE = \sum_{k=1}^S \frac{\sum_{i=1}^m (T_i^k - P_i^k)^2}{S} \quad (4.1)$$

Burada S eğitim örneklerinin sayısıdır. T_i^k , i girişinin gerçek çıktısıdır (k eğitim örneği girişte olduğunda) ve P_i^k , i girişinin öngörülen çıkış değeridir (k eğitim örneği kullanıldığında).

Min-Max normalizasyonu, veri setinin aralığının belirli aralık dâhilinde olduğu veriler için doğrusal bir dönüşüm yöntemidir. Öznitelikler 0 ile 1 veya -1 ile 1 aralığına düşecek şekilde yeniden ölçeklenmektedir. Bu yöntem genellikle denklem 4.2 kullanılarak hesaplanır:

$$V' = \frac{V - \min_A}{\max_A - \min_A} (\text{yeni_max}_A - \text{yeni_min}_A) + \text{yeni_min}_A \quad (4.2)$$

Normalizasyon, farklı aralıklardaki özniteliklere sahip veri kümelerini çözerken MLP için önemli bir aşamadır. Tezde, veri setleri min-max normalizasyonu kullanılarak ilk önce [0,1] aralığına dönüştürülmüştür, sonra eğitim ve test olarak iki gruba bölünmüştür. GWO, NN'yi amaç fonksiyonu olarak kullanmaktadır ve NN, eğitim verilerini sınıflandırdıktan sonra ortalama MSE'yi GWO'ya geri döndürmektedir. GWO algoritması en iyi ağırlıkları seçmektedir, sonrasında NN, Şekil 4.1'de gösterildiği gibi bu ağırlıkları kullanmıştır.

4.1 Deneyisel Çalışmalar

Tez kapsamında önerilen GWO-NN hibrit algoritmanın sonuçları; Naïve Bayes (NB), geri yayımlı çok katmanlı algılayıcı (Multi-Layer Perceptron with Back

propagation, MLP-BP), NN ile Parçacık Sürü Optimizasyonu (PSO-NN) ve NN ile Gravitasyonel Arama Algoritması (GSA-NN) sonuçları ile karşılaştırılmıştır. Kullanılan NN'nin yapısı $2 \times N + 1$ 'dir (Mirjalili, 2015).

Optimizasyon algoritmalarının popülasyon boyutu (10, 20, 30, 50, 75 ve 100) ve maksimum iterasyon (100, 150, 200 ve 250) olarak seçilmiştir. Algoritmaların her biri için 15 defa uygulama çalıştırılarak testler gerçekleştirilmiştir. NSL-KDD veri seti için elde edilen sonuçlar Tablo 4.1 ve Şekil 4.2 de gösterilmiştir. NSL-KDD veri setinde orijinalinde olduğu gibi eğitim için 125973 ve test için 22544 örneklem ayrımı tercih edilmiştir.

Tablo 4.1. NSL-KDD veri setinin sonuçları

Algoritmalar	Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
NB	88.76 %	3.3678e-02 $\pm$ 1.5280e-03
MLP-BP	96.00 %	2.0067e-02 $\pm$ 5.0279e-04
PSO-NN	91.34 %	6.923e-05 $\pm$ 7.615e-05
GSA-NN	81.45 %	5.361e-03 $\pm$ 4.659e-03
GWO-NN	97.83 %	2.669e-05 $\pm$ 2.311e-05

Popülasyon büyüklüğünün (10, 20, 30, 50, 75 ve 100) değerlerinden ve iterasyon sayılarında (100, 150, 200 ve 250) değerlerinden seçilerek üretilmiş doğruluk ve MSE değerleri sırasıyla Tablo 4.2, Tablo 4.3 ve Tablo 4.4'te gösterilmektedir.

Tablo 4.2. NSL-KDD veri seti kullanılarak elde edilen PSO-NN sınıflandırma sonuçları

Popülasyon	İterasyon Sayısı= 250		İterasyon Sayısı= 200	
	Doğruluk	MSE (AVG $\pm$ STD)	Doğruluk	MSE (AVG $\pm$ STD)
10	77.13	2.407e-02 $\pm$ 1.125e-02	78.48	3.099e-02 $\pm$ 2.637e-02
20	83.80	1.190e-03 $\pm$ 4.748e-04	74.13	7.535e-04 $\pm$ 2.375e-04
30	76.10	1.919e-03 $\pm$ 3.799e-04	84.24	4.575e-04 $\pm$ 6.471e-04
50	87.21	3.239e-04 $\pm$ 4.573e-04	88.59	2.985e-09 $\pm$ 9.747e-10
75	83.68	3.335e-06 $\pm$ 4.043e-06	88.34	1.714e-08 $\pm$ 4.246e-11
100	91.34	6.923e-05 $\pm$ 7.615e-05	89.15	4.246e-11 $\pm$ 6.004e-11
	İterasyon Sayısı= 150		İterasyon Sayısı= 100	
	Doğruluk	MSE (AVG $\pm$ STD)	Doğruluk	MSE (AVG $\pm$ STD)
10	77.67	9.179e-03 $\pm$ 8.155e-03	77.38	2.473e-02 $\pm$ 1.802e-03
20	82.59	7.847e-04 $\pm$ 1.110e-03	75.97	2.546e-02 $\pm$ 3.941e-03
30	84.24	4.575e-04 $\pm$ 6.471e-04	87.32	9.363e-03 $\pm$ 1.595e-03

50	86.00	2.985e-06 ± 9.747e-7	88.40	1.278e-02 ± 2.785e-03
75	80.34	1.732e-08 ± 2.398e-08	87.45	1.101e-02 ± 3.370e-03
100	86.24	2.746e-08 ± 3.875e-08	88.65	8.768e-03 ± 3.783e-03

Tablo 4.3. NSL-KDD veri seti kullanılarak elde edilen GSA-NN sınıflandırma sonuçları

Popülasyon	İterasyon Sayısı= 250		İterasyon Sayısı= 200	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	67.36	3.414e-01 ± 9.763e-02	69.52	3.120e-02 ± 4.508e-03
20	72.25	1.689e-02 ± 1.072e-02	71.41	1.950e-02 ± 5.023e-03
30	77.48	1.487e-02 ± 3.787e-03	73.91	1.083e-02 ± 2.353e-03
50	78.15	1.407e-02 ± 6.334e-03	73.98	8.086e-03 ± 1.700e-03
75	80.71	1.040e-02 ± 5.546e-03	75.09	1.074e-02 ± 2.799e-03
100	81.45	5.361e-03 ± 4.659e-03	81.25	6.218e-03 ± 2.471e-03
	İterasyon Sayısı= 150		İterasyon Sayısı= 100	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	71.52	3.477e-02 ± 3.303e-02	67.38	2.473e-02 ± 1.594e-02
20	72.86	2.206e-02 ± 2.747e-02	65.97	2.546e-02 ± 1.808e-02
30	72.81	1.412e-02 ± 6.788e-03	77.32	9.913e-03 ± 2.231e-03
50	72.93	8.121e-03 ± 2.522e-03	77.53	2.778e-02 ± 4.286e-03
75	75.37	2.777e-03 ± 1.174e-03	77.29	1.596e-02 ± 1.193e-02
100	77.09	5.356e-03 ± 2.157e-03	78.68	7.768e-03 ± 3.783e-03

Tablo 4.4. NSL-KDD veri seti kullanılarak elde edilen GWO-NN sınıflandırma sonuçları

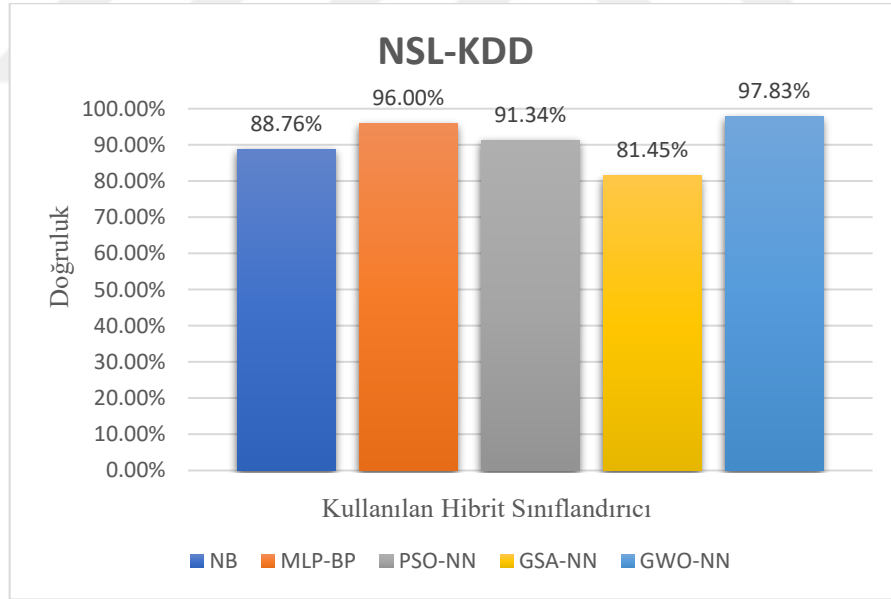
Popülasyon	İterasyon sayısı=250		İterasyon sayısı=200	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	94.26	9.363e-03 ± 6.830e-03	95.19	4.935e-03 ± 3.053e-03
20	94.72	8.228e-03 ± 5.163e-03	94.89	1.415e-03 ± 1.149e-03
30	96.12	0.005365 ± 0.004644	95.39	0.002613 ± 0.002098
50	95.26	4.003e-03 ± 5.291e-03	97.06	2.005e-03 ± 3.464e-03
75	96.79	1.998e-03 ± 2.275e-05	96.19	4.678e-03 ± 1.157e-03
100	97.83	2.669e-05 ± 2.311e-05	97.45	2.003e-04 ± 1.997e-04
	İterasyon sayısı=150		İterasyon sayısı=100	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	94.66	6.109e-03 ± 3.497e-03	95.12	6.142e-03 ± 3.874e-03
20	94.89	9.543e-03 ± 5.967e-03	95.92	1.419e-03 ± 2.256e-03
30	95.39	0.001321 ± 0.000904	95.46	0.002002 ± 0.002001
50	96.52	5.336e-03 ± 4.163e-03	96.12	1.349e-03 ± 2.319e-03
75	95.98	2.000e-03 ± 2.000e-03	95.52	2.673e-03 ± 3.047e-03
100	96.20	2.019e-04 ± 1.982e-04	95.87	4.000e-03 ± 3.464e-03

Tablo 4.5. NSL-KDD veri seti kullanılarak elde edilen NB sınıflandırma sonuçları

Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
87.52 %	1.2871e-01 $\pm$ 1.9184e-01
86.05 %	2.0741e-01 $\pm$ 2.1156e-01
88.76 %	3.3678e-02 $\pm$ 1.5280e-03
87.07 %	2.2408e-01 $\pm$ 2.1939e-01

Tablo 4.6. NSL-KDD veri seti kullanılarak elde edilen MLP-BP sınıflandırma sonuçları

Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
95.75 %	1.3470e-03 $\pm$ 1.3534e-03
96.00 %	2.0067e-02 $\pm$ 5.0279e-04
95.06 %	1.1209e-02 $\pm$ 1.1128e-02
93.08 %	6.3126e-02 $\pm$ 1.2585e-01

**Şekil 4.2.** NSL-KDD veri seti kullanılarak elde edilen Algoritma Doğruluk Grafikleri

NSL-KDD veri seti için Tablo 4.1 ve Şekil 4.2 de sunulan sonuçlar incelendiğinde, önerilen GWO-NN algoritmasının % 97,83 doğruluk değerini elde ederek en iyi doğruluğa ulaştığı görülmüştür. Ayrıca önerilen algoritmanın yerel optimumdan kaçınabildiği de görülmüştür. MLP-BP ise %96,00 doğruluk değeri ile en iyi ikinci

doğruluğa sahip olmuştur ancak yerel optimuma takıldığı görülmektedir. Diğer algoritmalar PSO-NN, NB ve GSA-NN ise sırasıyla %91.34, %88.76 ve %81.45 lik doğruluk değerlerini elde etmişlerdir.

UNSW-NB15 veri seti üzerinde de deneysel çalışmalar aynı şartlarda gerçekleştirilmiş ve elde edilen sonuçlar Tablo 4.7 ve Şekil 4.3 de sunulmuştur. UNSW-NB15 veri setinde orijinalinde olduğu gibi eğitim için 175341 ve test için 82332 örneklem ayrımı tercih edilmiştir. Tablo 4.5 ve Şekil 4.3 incelediğinde UNSW-NB15 veri seti için, GWO-NN en iyi doğruluk ve MSE ($1.51e-13 \pm 9.14e-14$) değerine sahip olup yerel optimuma takılmamıştır. PSO-NN %97.60'lık en iyi ikinci doğruluğu elde etmiş ve MSE'de iyi değerlerine ulaşarak yerel optimumdan kurtulduğu görülmüştür. MLP-BP ise %97.50 değeri ile iyi bir doğruluk elde etmiş, ancak yine de yerel optimuma takılmayı engelleyememiştir. Diğer iki algoritma olan NB ve GSA-NN ise en düşük doğruluğa sahip olmuştur.

Tablo 4.7. UNSW-NB15 veri setine göre Algoritmaların sonuçları

Algoritmalar	Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
NB	92.88 %	$2.1341e-02 \pm 6.7436e-03$
MLP-BP	97.50 %	$1.3002e-04 \pm 1.9670e-04$
PSO-NN	97.60 %	$6.128e-05 \pm 4.858e-06$
GSA-NN	82.63 %	$1.294e-04 \pm 2.314e-05$
GWO-NN	98.21 %	$1.51e-13 \pm 9.14e-14$

Bu deneyde kullanılan Popülasyon büyüklüğünün (10, 20, 30, 50, 75, 85 ve 100) değerlerinden ve iterasyon sayılarının (100, 150, 200 ve 250) değerlerinden seçilerek üretilmiş doğruluk ve MSE değerleri sırasıyla Tablo 4.8, Tablo 4.9 ve Tablo 4.10'da gösterilmektedir.

Tablo 4.8. UNSW-NB15 veri seti kullanılarak elde edilen PSO-NN sınıflandırıcı sonuçları

Popülasyon	İterasyon sayısı= 250		İterasyon sayısı=200	
	Doğruluk	MSE (AVG $\pm$ STD)	Doğruluk	MSE (AVG $\pm$ STD)
10	77.62	$3.414e-02 \pm 9.763e-03$	79.52	$3.120e-02 \pm 4.508e-03$
20	82.25	$1.689e-02 \pm 1.072e-02$	81.41	$1.950e-02 \pm 5.023e-03$
30	87.48	$1.392e-02 \pm 4.622e-03$	87.64	$1.083e-02 \pm 2.353e-03$

50	88.15	1.057e-02 ± 1.073e-03	89.98	1.043e-02 ± 3.746e-03
75	90.70	1.103e-04 ± 5.568e-03	88.42	1.074e-02 ± 2.799e-03
100	97.60	6.128e-05 ± 4.858e-06	92.26	4.335e-03 ± 2.762e-03
		İterasyon sayısı=150	İterasyon sayısı=100	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	81.52	3.810e-02 ± 1.979e-02	85.14	2.306e-02 ± 1.202e-02
20	82.86	1.739e-02 ± 1.521e-02	85.64	1.932e-02 ± 5.012e-03
30	86.14	1.328e-02 ± 5.003e-03	86.31	1.156e-02 ± 4.145e-03
50	91.81	7.787e-03 ± 3.442e-03	90.74	6.389e-03 ± 1.496e-03
75	92.04	3.111e-03 ± 1.037e-03	94.07	7.415e-03 ± 1.664e-03
100	92.20	5.356e-03 ± 2.541e-04	94.99	2.323e-03 ± 1.535e-04

Tablo 4.9. UNSW-NB15 veri seti kullanılarak elde edilen GSA-NN sınıflandırıcı sonuçları

Popülasyon	İterasyon sayısı=250		İterasyon sayısı=200	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	68.89	2.827e-02 ± 1.945e-02	68.38	4.084e-02 ± 1.202e-02
20	69.30	1.160e-03 ± 4.455e-04	68.69	7.285e-04 ± 2.304e-04
30	67.09	1.919e-03 ± 3.799e-04	74.30	3.070e-04 ± 4.342e-04
50	74.71	3.239e-04 ± 4.573e-04	77.59	2.375e-09 ± 1.668e-09
75	74.81	3.780e-06 ± 4.842e-06	77.82	1.677e-08 ± 2.335e-08
100	82.63	1.294e-04 ± 2.314e-05	77.90	2.335e-09 ± 3.182e-09
		İterasyon sayısı=150	İterasyon sayısı=100	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	67.77	8.279e-03 ± 6.599e-03	70.21	1.706e-02 ± 7.777e-03
20	72.67	1.285e-03 ± 1.817e-03	70.66	1.982e-02 ± 2.891e-03
30	74.29	4.125e-04 ± 5.834e-04	71.31	1.151e-02 ± 2.801e-03
50	74.64	2.465e-09 ± 1.544e-10	72.02	6.389e-03 ± 1.496e-03
75	75.00	1.880e-08 ± 2.613e-08	74.80	5.215e-03 ± 5.992e-04
100	76.19	2.295e-08 ± 3.239e-08	78.01	2.323e-03 ± 1.535e-03

Tablo 4.10. UNSW-NB15 veri seti kullanılarak elde edilen GWO-NN sınıflandırıcı sonuçları

Popülasyon	İterasyon sayısı=250		İterasyon sayısı=200	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)
10	95.64	1.81e-06 ± 1.59e-06	90.00	4.34e-05 ± 8.27e-05
20	92.75	3.64e-07 ± 5.07e-07	91.89	2.07e-06 ± 2.64e-06
30	94.22	7.35e-10 ± 4.63e-10	91.44	8.00e07 ± 1.80e-06
50	94.77	2.82e-11 ± 2.23e-11	93.11	2.07e-06 ± 2.64e-06
75	93.14	3.14e-08 ± 4.43e-08	93.67	2.44e-09 ± 3.44e-09
100	98.21	1.51e-13 ± 9.14e-14	94.11	7.60e-10 ± 1.07e-09
		İterasyon sayısı=150	İterasyon sayısı=100	
	Doğruluk	MSE (AVG ± STD)	Doğruluk	MSE (AVG ± STD)

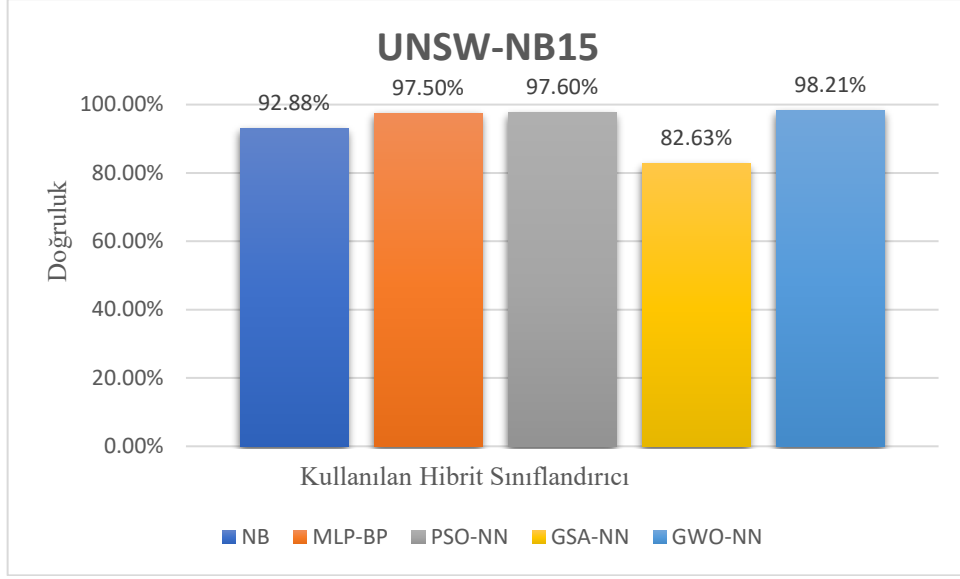
10	86.00	0.002688 ± 0.00376	93.05	0.0109553 ± 0.0244457
20	91.55	$1.93e-04 \pm 3.25e-04$	96.83	0.0105533 ± 0.02575677
30	92.10	$4.75e-08 \pm 8.40e-08$	92.94	0.0016871 ± 0.0034005
50	96.12	$1.33e-09 \pm 1.55e-09$	97.39	$4.364e-09 \pm 6.171e-09$
75	95.63	$1.79e-04 \pm 2.53e-04$	94.75	0.0003969 ± 0.001179
100	92.00	$1.44e-11 \pm 2.51e-12$	90.11	0.0087916 ± 0.0244226

Tablo 4.11. UNSW-NB15 veri seti kullanılarak elde edilen NB sınıflandırma sonuçları

Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
92.00 %	$4.8007e-02 \pm 2.7848e-02$
92.22 %	$3.2304e-02 \pm 3.9046e-02$
91.52 %	$4.4674e-02 \pm 4.7071e-02$
92.88 %	$2.1341e-02 \pm 6.7436e-03$

Tablo 4.12. UNSW-NB15 veri seti kullanılarak elde edilen MLP-BP sınıflandırma sonuçları

Doğruluk	Ortalama Kare Hatası (AVG $\pm$ STD)
97.50 %	$1.3002e-04 \pm 1.9670e-04$
96.59 %	$1.2013e-03 \pm 2.0522e-03$
95.38 %	$1.2391e-03 \pm 2.0202e-03$
96.93 %	$8.0256e-04 \pm 1.0837e-03$



Şekil 4.3. UNSW-NB15 veri seti kullanılarak elde edilen Algoritma Doğruluk Grafikleri

Tablo 4.13. Hata Matrisini

		Tahmin edilen	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Negatif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Doğruluk (Accuracy - ACC), bir sınıflandırıcı tarafından doğru şekilde sınıflandırılan örnek sayısının toplam örnek sayısına bölünmesiyle elde edilir; doğruluk değerlendirmesi, test seti kullanılarak hesaplanır, en iyi doğruluk 1'dir ve en kötü doğruluk ise 0'dır.

$$\text{Doğruluk}(ACC) = \frac{DP + DN}{DP + DN + YP + YN} \quad (4.3)$$

Duyarlılık veya Doğru Pozitif Oranı(True Positive Rate - TPR), pozitif olarak sınıflandırılan örnek sayısının tüm pozitif örneklere bölünmesiyle elde edilen hesaplamaadır.

$$\text{Duyarlılık}(TPR) = \frac{DP}{DP + YN} \quad (4.4)$$

Özgüllük/Seçicilik veya Doğru Negatif Oranı(True Negative Rate - TNR) , negatif bir değer olarak sınıflandırılan örneklerin oranının, gerçek değeri negatif olan tüm örneklerin sayısına bölünmesidir.

$$Seçicilik(TNR) = \frac{DN}{DN + YP} \quad (4.5)$$

Kesinlik veya Pozitif Öngörü Değeri(Positive Predictive Value - PPV), pozitif bir değer olarak sınıflandırılan örnek sayısının, pozitif bir değere sınıflandırılan tüm örneklerle bölünmesiyle elde edilen hesaplamadır.

$$Kesinlik(PPV) = \frac{DP}{DP + YP} \quad (4.6)$$

F-Skoru, kesinlik ve duyarlılığın harmonik ortalamasıdır, bu nedenle her iki kriteri birlikte değerlendirmek daha doğrudur çünkü doğruluk ve duyarlılık kriterleri tek başına anlamlı bir karşılaştırma sonucu elde etmek için yeterli değildir.

$$F - Skor = 2 * \frac{Kesinlik * Duyarlilik}{Kesinlik + Duyarlilik} \quad (4.7)$$

Tablo 4.14. NSL-KDD veri seti için GWO-NN yönteminin sınıf hataları (popülasyon=100, iterasyon sayısı=250 alınmıştır)

	GWO-NN
Doğruluk	97.83 %
Duyarlılık	98.13 %
Özgüllük	97.22 %
Kesinlik	98.43 %
F-Skoru	98.28 %
Hata Oranı	2.17 %

NSL-KDD veri seti için önerilen yöntemin performans ölçümüne göre doğruluk, duyarlılık, özgüllük, kesinlik, f-skor ve hata oranı hesaplanmış ve sonuçlar sırasıyla tablo 4.14'te gösterilmiştir.

Tablo 4.15. UNSW-NB15 veri seti için GWO-NN yönteminin sınıf hataları (popülasyon=100, iterasyon sayısı=250 alınmıştır)

	GWO-NN
Doğruluk	98.21 %
Duyarlılık	98.84 %
Özgüllük	96.77 %
Kesinlik	98.55 %
F-Skoru	98.69 %
Hata Oranı	1.79 %

UNSW-NB15 veri seti için önerilen yöntemin performans ölçümüne göre doğruluk, duyarlılık, özgüllük, kesinlik, f-skor ve hata oranı hesaplanmış ve sonuçlar sırasıyla tablo 4.15'te gösterilmiştir.

Tez kapsamında gerçekleştirilen deneylerde, NB eğitim sırasında diğer algoritmalar arasında en hızlı algoritma olmuştur. Her iki veri kümesinin doğruluk ve MSE değerleri karşılaştırıldığında, UNSW-NB15 veri kümesi için yapılan deneylerde elde edilen sonuçların NSL-KDD veri kümesinden elde edilenlere göre daha iyi olduğu görülmüştür. UNSW-NB15 veri setinin bulut bilişim güvenliği için saldırı tespit verisi olarak kullanabileceği de gözlemlenmiştir. Ayrıca, deneysel sonuçlar GWO ve NN kombinasyonunun bulut üzerindeki saldırıları iyi bir performansla tespit edebileceğini göstermiştir.

5. SONUÇ

Bulut Bilişim; verileri ve uygulamaları kullanmak, depolamak ve işlemek için internet bağlantısına ve uzak merkezi sunuculara dayanan ve kullanıcıların daha yüksek bir hızda daha düşük bir fiyata verilere erişmelerini sağlayan hizmet tabanlı bir teknolojidir. Yeni nesil bir bilgi teknolojisi olan bulut bilişimin literatürde birçok tanımı vardır, ancak ilgili kaynaklar tarafından sıklıkla belirtilen ve NIST tarafından tanınan bulut istihbaratları, yapılandırılabilir veri ortak koleksiyonlarına ve IT kaynakları (bilgisayar ağları, veri depolama, sunucular, hizmetler ve uygulamalar vb.) gibi her zaman, uygun bir şekilde ve her durumda, herhangi bir zamanda, herhangi bir yerde erişim sağlayan bir modeldir.

Bulut bilişim üzerinden sunulan hizmetler, Hizmet Olarak Her Şey (XaaS) olarak tanımlanır ve XaaS, Hizmet Olarak İzleme (MaaS), Hizmet Olarak Ağ (NaaS) veya Hizmet Olarak İletişim (CaaS) gibi hizmetleri içerebilir. Ancak, bulut bilişimin temel hizmetleri Hizmet Olarak Altyapı (IaaS), Hizmet Olarak Yazılım (SaaS) ve Hizmet Olarak Platform (PaaS)'dur.

Bulut bilişimde karşılaşılan güvenlik riskleri, veri gizliliği ve gizliliği koruması, yönetim yetersizliği, yönetim arayüzünde olası güvenlik açığı, bulut çalışanlarının kötü niyetli davranışları, kullanılabilirlik garantisi, izolasyon hatası, uyumluluk ve yasal riskler olarak belirtilmiştir.

Bir saldırı, herhangi bir işlem dizisi veya bir bilgisayar kaynağının bütünlüğünü, gizliliğini ve erişilebilirliğini tehlikeye atan eylemler olarak tanımlanabilir. Saldırı tespit sistemleri, bilgi güvenliği fikrinin erken aşamalarında, saldırı kavramının ortaya çıkması ve sistem güvenliği için paha biçilmez bir unsur olacak saldırıları otomatik olarak tespit etmek için bir mekanizmaya ihtiyaç duyulmasıyla ortaya çıkmıştır. Saldırı tespiti ilk aşamada çok başarılı sonuçlar elde edemezken, yeni teknoloji ve bilgi altyapısında köklü bir değişiklik olmadığı sürece geliştirilecek ve kullanılacak bir sistem olmaya devam etmektedir. Saldırı Tespiti, bir ağda veya sistemde meydana gelen trafiği ve olayları sürekli izleyen, analiz eden ve daha sonra sisteme veya ağa giren paketleri bir sonuç olarak kontrol ederek zararlı olanları tespit eden süreçtir. IDS'nin gözlemlediği sistem; bir ağ, bilgisayar veya herhangi bir bilgi sistemi kaynağı olabilir. Günümüzde, çeşitli algoritmalar kullanarak geliştirmekte olan sistemler, yüksek sayıda saldırıyı önleyen ve saldırganları yavaşlatan bilgi güvenliğinin temel taşlarından biridir.

İnternetin yaygın kullanımı ile birlikte, bilişim sistemleri ve saldırı türlerindeki genişleme ile güvenlik tehditlerinde önemli artışlar olmuştur. Tehditler ve saldırılar nedeniyle yeni mekanizmaların geliştirilmesi gerekliliği ortaya çıkmıştır. Bulut bilişim, her gün birçok saldırı ile karşı karşıyadır. Bu tezde, Gray-kurt Optimasyonu ve Yapay sinir ağı algoritması bulut güvenliğini arttırmak için hibrit olarak değerlendirilmiştir. Sonuçlar, GWO-NN 'nin bulut üzerindeki izinsiz girişleri tespit edebildiğini kanıtlamıştır. Önerilen hibrit algoritmanın UNSW-NB15 veri seti için NSL-KDD veri setine kıyasla üstün sonuçlar sergilediği gösterilmiştir.

Tezden sonraki süreçte, kullanılan veri setleri zenginleştirilerek algoritmanın başarısı ve performansı izlenecektir. Ayrıca, diğer makine öğrenmesi algoritmaları ile hibritlemeler gerçekleştirilerek daha melez veya daha başarılı algoritmalar ortaya konulabilir.

KAYNAKLAR

- Al Haddad, Z., Hanoune, M., & Mamouni, A. (2016). *A collaborative framework for intrusion detection (C-NIDS) in Cloud computing*. Paper presented at the 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech).
- Alharkan, T., & Martin, P. (2012). *IDSaaS: Intrusion detection system as a service in public clouds*. Paper presented at the Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012).
- Alqahtani, S. M., & John, R. (2017). *A comparative analysis of different classification techniques for cloud intrusion detection systems' alerts and fuzzy classifiers*. Paper presented at the 2017 Computing Conference.
- Alzubi, Q. M., Anbar, M., Alqattan, Z. N., Al-Betar, M. A., & Abdullah, R. (2019). Intrusion detection system based on a modified binary grey wolf optimisation. *Neural Computing and Applications*, 1-13.
- Arjunan, K., & Modi, C. N. (2017). *An enhanced intrusion detection framework for securing network layer of cloud computing*. Paper presented at the 2017 ISEA Asia Security and Privacy (ISEASP).
- Avdagic, I., & Hajdarevic, K. (2017). *Survey on machine learning algorithms as cloud service for CIDPS*. Paper presented at the 2017 25th Telecommunication Forum (TELFOR).
- Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, 13(4), 27-31.
- Bhamare, D., Salman, T., Samaka, M., Erbad, A., & Jain, R. (2016). *Feasibility of supervised machine learning for cloud security*. Paper presented at the 2016 International Conference on Information Science and Security (ICISS).
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3), 268-308.
- Canlı, C. (2009). Saldırı Tesbit Sistemlerinin İncelemesi ve Bu Bağlamda Bilgi Güvenliği. *Masters Degree*.
- Cisco Global Cloud Index. (2018). Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html#_Toc503317519 [Ziyaret Tarihi: 12 Temmuz 2020]
- Cliff, D. (2010). Remotely hosted services and 'cloud computing'.
- Davis, L. 1991, Handbook of genetic algorithms. In: Van Nostrand Reinhold, New York.
- Derfouf, M., & Eleuldj, M. (2017). *Performance analysis of intrusion detection systems in the cloud computing*. Paper presented at the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech).
- Dorffner, G. (1996). *Neural networks for time series processing*. Paper presented at the Neural network world.
- Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1), 53-66.
- Duman, E., Uysal, M., & Alkaya, A. F. (2012). Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences*, 217, 65-77.

- Eberhart, R., & Kennedy, J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science.
- Elhariri, E., El-Bendary, N., Hassanien, A., & Abraham, A. (2015). Grey wolf optimization for one-against-one multi-class support vector machines. *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 7-12.
- ERGİN, O., Gerede, Ç., & Korkmaz, Y. (2012). Kamuda Bulut Bilişim, 1. *Çalışma Grubu, Türkiye Bilişim Derneği Kamu Bilgi İşlem Merkezleri Yöneticileri Birliği, Ankara, Nisan*.
- Furht, B., & Escalante, A. (2010). *Handbook of cloud computing* (Vol. 3): Springer.
- Ghosh-Dastidar, S., & Adeli, H. (2009). Spiking neural networks. *International journal of neural systems*, 19(04), 295-308.
- Gupta, S., & Kumar, P. (2015). An immediate system call sequence based approach for detecting malicious program executions in cloud environment. *Wireless Personal Communications*, 81(1), 405-425.
- Hajimirzaei, B., & Navimipour, N. J. (2019). Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. *ICT Express*, 5(1), 56-59.
- He, Z., Zhang, T., & Lee, R. B. (2017). *Machine learning based DDoS attack detection from source side in cloud*. Paper presented at the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud).
- Hertz, A., & Widmer, M. (2003). Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 151(2), 247-252.
- Hertz, J. A., Krogh, A. S., & Palmer, R. G. Introduction to the Theory of Neural Computation. In: Citeseer.
- Hogan, M., Liu, F., Sokol, A., & Tong, J. (2011). Nist cloud computing standards roadmap. *NIST Special Publication*, 35, 6-11.
- Idhammad, M., Afdel, K., & Belouch, M. (2018). Distributed intrusion detection system for cloud environments based on data mining techniques. *Procedia Computer Science*, 127, 35-41.
- Inayat, Z., Gani, A., Anuar, N. B., Anwar, S., & Khan, M. K. (2017). Cloud-based intrusion detection and response system: open research issues, and solutions. *Arabian Journal for Science and Engineering*, 42(2), 399-423.
- Javadpour, A., Abharian, S. K., & Wang, G. (2017). *Feature selection and intrusion detection in cloud environment based on machine learning algorithms*. Paper presented at the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC).
- Kannan, A., Maguire Jr, G. Q., Sharma, A., & Schoo, P. (2012). *Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks*. Paper presented at the 2012 IEEE 12th International Conference on Data Mining Workshops.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Retrieved from
- Kholidy, H. A., & Baiardi, F. (2012a). *Cidd: A cloud intrusion detection dataset for cloud computing and masquerade attacks*. Paper presented at the 2012 Ninth International Conference on Information Technology-New Generations.

- Kholidy, H. A., & Baiardi, F. (2012b). *CIDS: A framework for intrusion detection in cloud systems*. Paper presented at the 2012 Ninth International Conference on Information Technology-New Generations.
- Kohonen, T., Oja, E., Simula, O., Visa, A., & Kangas, J. (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10), 1358-1384.
- Kong, J. (2010). *Protecting the confidentiality of virtual machines against untrusted host*. Paper presented at the 2010 International Symposium on Intelligence Information Processing and Trusted Computing.
- Kong, J. (2011). AdjointVM: A new intrusion detection model for cloud computing. *Energy Procedia*(13), 7902-7911.
- Kumar, A., Kumar, V., Singh, P., & Kumar, A. (2012). A Novel approach: Security measures and Concerns of Cloud Computing. *International Journal of Computer Technology and Applications*, 3(3), 1008-1014.
- Kwasniewski, T. J., & Puig, E. (2011). Cloud computing in the government. *Data & Analysis Center for Software*.
- Leavitt, N. (2009). Is cloud computing really ready for prime time? *Computer*(1), 15-20.
- Li, Z., Sun, W., & Wang, L. (2012). *A neural network based distributed intrusion detection system on cloud platform*. Paper presented at the 2012 IEEE 2nd international conference on Cloud Computing and Intelligence Systems.
- Ling, C.-H., Hsien, W.-F., & Hwang, M.-S. (2016). A Double Circular Chain Intrusion Detection for Cloud Computing Based on AdjointVM Approach. *IJ Network Security*, 18(2), 397-400.
- Lonea, A. M., Popescu, D. E., & Tianfield, H. (2013). Detecting DDoS attacks in cloud computing environment. *International Journal of Computers Communications & Control*, 8(1), 70-78.
- Marinova-Boncheva, V. (2007). A short survey of intrusion detection systems. *problems of Engineering Cybernetics and Robotics*, 58, 23-30.
- Marwan, M., Kartit, A., & Ouahmane, H. (2018). Security Enhancement in Healthcare Cloud using Machine Learning. *Procedia Computer Science*, 127, 388-397.
- McCulloch, W. S., & Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, 52(1-2), 99-115.
- Mehmood, Y., Shibli, M. A., Habiba, U., & Masood, R. (2013). *Intrusion detection system in cloud computing: Challenges and opportunities*. Paper presented at the 2013 2nd National Conference on Information Assurance (NCIA).
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1), 150-161.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, 36(1), 42-57.
- Moustafa, N., & Slay, J. (2015). *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. Paper presented at the 2015 military communications and information systems conference (MilCIS).
- NUR, I. M., & ÜLKER, E. (2018). A hybrid cloud-based Intrusion Detection and Response System (IDRS) based on Grey Wolf Optimizer (GWO) and Neural Network (NN).

- Oktaý, U., Aydin, M. A., & Sahingoz, O. K. (2013). *A circular chain intrusion detection for cloud computing based on improved AdjointVM approach*. Paper presented at the 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI).
- Oktaý, U., & Sahingoz, O. (2013). *Attack types and intrusion detection systems in cloud computing*. Paper presented at the Proceedings of the 6th International Information Security & Cryptology Conference.
- Pan, W.-T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69-74.
- Park, J., & Sandberg, I. W. (1993). Approximation and radial-basis-function networks. *Neural computation*, 5(2), 305-316.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, 179(13), 2232-2248.
- Rimal, B. P., Jukan, A., Katsaros, D., & Goeleven, Y. (2011). Architectural requirements for cloud computing systems: an enterprise cloud approach. *Journal of Grid Computing*, 9(1), 3-26.
- Scarfone, K., & Mell, P. (2012). *Guide to intrusion detection and prevention systems (idps)*. Retrieved from
- Tarantilis, C. D., Ioannou, G., & Prastacos, G. (2005). Advanced vehicle routing algorithms for complex operations management problems. *Journal of Food Engineering*, 70(3), 455-471.
- Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set*. Paper presented at the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.
- Trusted Computing Group. (2014). Retrieved from <http://www.trustedcomputinggroup.org/> [Ziyaret Tarihi: 18 Haziran 2020]
- Tyler, G. (2009). Information Assurance Tools Report Intrusion Detection Systems. *Information Assurance Technology Analysis Center (IATAC)*.
- ÜLKER, E., & NUR, I. M. (2020). A Novel Hybrid IoT Based IDS Using Binary Grey Wolf Optimizer (BGWO) and Naive Bayes (NB). *Avrupa Bilim ve Teknoloji Dergisi*, 279-286.
- Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, 153-171.
- Wang, G.-G., Gandomi, A. H., Alavi, A. H., & Hao, G.-S. (2014). Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Computing and Applications*, 25(2), 297-308.
- Wang, G.-G., Guo, L., Gandomi, A. H., Hao, G.-S., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences*, 274, 17-34.
- Yang, X.-S. (2010a). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209-218): Springer.
- Yang, X.-S. (2010b). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.
- Youssef, B. C., Nada, M., Elmehdi, B., & Boubker, R. (2016). *Intrusion detection in cloud computing based attacks patterns and risk assessment*. Paper presented at the 2016 Third International Conference on Systems of Collaboration (SysCo).
- YÜKSEL, H. (2012). BULUT BİLİŞİM EL KİTABI. 13-15. Retrieved from <https://yuksele.files.wordpress.com/2012/01/bulutbilic59fimekitabc4b1.pdf> sayfa 13-15 [Ziyaret Tarihi: 22 Mayıs 2020]
- Zekri, M., El Kafhali, S., Aboutabit, N., & Saadi, Y. (2017). *DDoS attack detection using machine learning techniques in cloud computing environments*. Paper

presented at the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech).

