T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ EKOSİSTEMİNDE YAPAY ZEKA TABANLI SALDIRI TESPİT SİSTEMİ GELİŞTİRİLMESİ

Umur KURİŞ

DANIŞMAN
Dr. Öğr. Üyesi Özgür Can TURNA

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

İSTANBUL-2020

Bu çalışma 30.12.2019 Tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı Yüksek Lisans Tezi olarak kabul edilmiştir.

TEZ JÜRİSİ

Dr.Öğr.Üyesi Özgür Can TURNA
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Muhammed Ali AYDIN
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Dr.Öğr.Üyesi Ali BOYACI
İstanbul Ticaret Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Bu tez çalışmasında Nesnelerin İnterneti Ekosisteminde Yapay Zeka Tabanlı Saldırı Tespit Sistemi geliştirilmesi hedeflenmiştir.

Tez konusu seçiminde isteklerimi göz önünde bulundurup bana yardımcı olan hocam Doç. Dr. Muhammed Ali AYDIN'a, tez çalışmamın planlanmasında, araştırılmasında ve yürütülmesinde ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren danışman hocam Dr. Öğr. Üyesi Özgür Can TURNA'ya ve tüm eğitim hayatım boyunca benden maddi manevi desteklerini esirgemeyen her zaman yanımda olan aileme teşekkürlerimi bir borç bilirim.

Aralık 2019 Umur KURİŞ

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v**
ŞEKİL LİSTESİ ..... **viii**
TABLO LİSTESİ ..... **ix**
SİMGE VE KISALTMA LİSTESİ ..... **x**
ÖZET ..... **xi**
SUMMARY ..... **xii**
1. GİRİŞ **1**
2. GENEL KISIMLAR **2**
2.1. NESNELERİN İNTERNETİ (INTERNET OF THINGS-IOT) 2
2.1.1. IoT Ekosisteminde Yer Alan Temel Kavramlar 2
2.1.1.1. Nesneler *2*
2.1.1.2. Sensörler *3*
2.1.1.3. Radyo Frekansı Tanımlama (RFID) *4*
2.1.1.4. Kontrolör *5*
2.1.1.5. Aktivatör *5*
2.1.1.6. Bulut Bilişim *6*
2.1.1.7. Sis Bilişim *7*
2.1.1.8. Büyük Veri *8*
2.1.2. IoT Mimarisi 9
2.1.3. IoT İçin Protokol Yığını 12
2.1.3.1. Uygulama Protokolleri *13*
2.1.3.2. Hizmet Bulma Protokolleri (Service Discovery Protocols-SDP) *14*
2.1.3.3. Altyapı Protokolleri *14*
2.1.3.4. Etkili Protokoller (Influential Protocols) *16*
2.1.3.5. Protokoller İçin Güvenlik Gereksinimleri *16*
2.1.4. IoT'de Güven Yönetimi 17
2.1.5. IoT İçin Mevcut Platformlar 20
2.2. SALDIRI TESPİT SİSTEMLERİ 21
2.2.1. Saldırı Tespitinin Tarihi 25

2.2.2. STS Gereksinimleri 26
2.2.3. STS'de Karar Verme 27
2.2.4. Saldırı Yanıtı 27
2.2.5. Saldırı Tespit Teknikleri 28
2.2.5.1. Anomali Tabanlı Saldırı Tespit Teknikleri 28
2.2.5.2. İmza Tabanlı Saldırı Tespit Teknikleri 28
2.2.6. STS'lerin Sınıflandırılması 29
2.2.6.1. Saldırgan Türüne Göre Sınıflandırma 30
2.2.6.2. Saldırı Türüne Göre Sınıflandırma 30
2.2.6.3. Tespit Metodolojilerine Göre Sınıflandırma 30
2.2.6.4. Denetlenen Verinin Kaynağına Göre Sınıflandırma 32
2.2.6.5. Toplanan Verilerin Hesaplanma Yerine Göre Sınıflandırma 32
2.2.6.6. Ağ Altyapılarına Göre Sınıflandırma 33
2.2.6.7. Kullanım Frekansına Göre Sınıflandırma 34
2.2.7. STS'lerin Avantajları ve Dezavantajları 34
3. MALZEME VE YÖNTEM 35
3.1. VERİ SETİ 35
3.2. KULLANILAN DİL VE KÜTÜPHANELER 36
3.3. MAKİNE ÖĞRENMESİ 36
3.3.1. Denetimli Öğrenme (Supervised Learning) 37
3.3.1.1. K-En Yakın Komşular (K-Nearest Neighbour-KNN) 37
3.3.1.2. Destek Vektör Makineleri (Support Vector Machine-SVM) 38
3.3.1.3. Karar Ağaçları (Decision Trees-DTs) 38
3.3.1.4. Rassal Ormanlar (Random Forest-RF) 39
3.3.1.5. Yapay Sinir Ağları (Artificial Neural Network-ANN) 40
3.3.1.6. Bayes Teoremi Tabanlı Algoritmalar 40
3.3.2. Denetimsiz Öğrenme (Unsupervised Learning) 41
3.3.2.1. Temel Bileşen Analizi (Principal Component Analysis-PCA) 41
3.3.2.2. K-Ortalamalar Kümeleme (K-Means Clustering) 41
3.3.2.3. Birliktelik Kuralı (Association Rule -AR) algoritmaları 42
3.3.3. Pekiştirmeli Öğrenme (Reinforcement Learning-RL) 43
4. BULGULAR 44
5. TARTIŞMA VE SONUÇ 48
KAYNAKLAR 50

ÖZGEÇMİŞ 60

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: IoT mimarisi. 9

Şekil 2.2: STS'lerin sınıflandırılması. 29

Şekil 3.1: Veri setinin ilk beş satırı. 35

Şekil 3.2: Veri setinin son beş satırı. 35

Şekil 4.1: K değerlerine karşılık gelen doğruluk oranları. 44

Şekil 4.2: KNN algoritması sınıflandırma sonuçları. 45

Şekil 4.3: DVM algoritması sınıflandırma sonuçları. 45

Şekil 4.4: Naive Bayes algoritması sınıflandırma sonuçları. 46

Şekil 4.5: Karar ağaçları algoritması sınıflandırma sonuçları. 46

Şekil 4.6: Rassal ormanlar algoritması sınıflandırma sonuçları. 47

Şekil 5.1: Uygulama ekran görüntüsü. 48

TABLO LİSTESİ

Sayfa No

Tablo 2.1: IoT protokol yığını. 12

Tablo 5.1: Algoritmalar, tahmin sonuçları ve doğruluk oranları. 48

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
DDoS	: Distrubuted Denial of Service (Dağıtık Servis Dışı Bırakma)
DoS	: Denial of Service (Servis Dışı Bırakma)
GUI	: Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
HIDS	: Host-based Intrusion Detection System (Bilgisayar Tabanlı Saldırı Tespit Sistemi)
HTTP	: Hypertext Transfer Protocol (Hipermetin Aktarma Prtokolü)
IDE	: Integrated Development Environment (Tümleşik Geliştirme Ortamı)
IDS	: Intrusion Detection System (Saldırı Tespit Sistemi)
IETF	: Internet Engineering Task Force (İnternet Mühendisliği Görev Grubu)
IoT	: Internet of Things (Nesnelerin İnterneti)
IP	: Internet Protocol
IPS	: Intrusion Prevention System (Saldırı Engelleme Sistemi)
KNN	: K Nearest Neighbour (K-En Yakın Komşular)
ML	: Machine Learning (Makine Öğrenmesi)
M2M	: Machine to Machine (Makineden Makineye)
NB	: Naive Bayes
NIDS	: Network-based Intrusion Detection System (Ağ Tabanlı Saldırı Tespit Sistemi)
QoS	: Quality of Servis (Hizmet Kalitesi)
RFID	: Radio Frequency Identification (Radyo Frekansı Tanımlama)
RL	: Reinforcement Learning (Pekiştirmeli Öğrenme)
SDK	: Software Development Kit (Yazılım Geliştirme Kiti)
SES	: Saldırı Engelleme Sistemi
STS	: Saldırı Tespit Sistemi
SVM	: Support Vector Machine (Destek Vektör Makinesi-DVM)
TCP	: Transmission Control Protocol(İletim Kontrol Protokolü)
WSN	: Wireless Sensor Network (Kablosuz Sensör Ağı)

ÖZET

NESNELERİN İNTERNETİ EKOSİSTEMİNDE YAPAY ZEKA TABANLI SALDIRI TESPİT SİSTEMİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Umur KURİŞ

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Özgür Can TURNA

Sayıları, çeşitleri ve kullanım alanları hızla artmakta olan akıllı cihazlar saldırganlar için ciddi bir hedef haline gelmiştir. Kullanım alanlarına bağlı olarak maruz kaldıkları saldırının tehlikesi de çok ciddi boyutlara erişebilecek olan akıllı cihazlar için üretim aşamasında artırılması gereken güvenlik önlemlerine ek olarak üretim sonrasında da güncel, tespit edilmemiş saldırıları engellemenin yolları araştırılmak zorundadır. Günümüzde; bilinmeyen, güncel saldırıları tespit etmek için makine öğrenmesi, bulanık mantık, yapay sinir ağları gibi yapay zeka alanlarına başvurulmaktadır.

Güncel teknoloji eğilimleri arasında yer alan nesnelerin interneti, yapay zeka ve siber güvenlik konularının bir arada araştırılmasını gerektiren bu tezin amacı; nesnelerin interneti ekosisteminde yer alan akıllı cihazların güvenliğini sağlayabilmek için bu cihazlara yönelik siber saldırıların tespit edilmesini sağlayabilecek yapay zeka uygulamalarının kullanılacağı anomali tabanlı bir saldırı tespit sisteminin geliştirilmesidir.

Aralık 2019, 72 sayfa.

Anahtar kelimeler: Güvenlik, Makine Öğrenmesi, Nesnelerin İnterneti

SUMMARY

DEVELOPING AN ARTIFICIAL INTELLIGENCE BASED INTRUSION DETECTION SYSTEM ON INTERNET OF THINGS ECOSYSTEM

M.Sc. THESIS

Umur KURİŞ

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Özgür Can TURNA

With their rapidly increasing numbers, varieties and applications, smart devices have become a serious target for attackers. Depending on their usage area, the danger of the attack may also be very serious for smart devices. Therefore, in addition to the security measures that need to be increased during the production phase for these devices; ways to prevent current, undetected attacks after production have to be investigated. Nowadays to detect unknown, current attacks, artificial intelligence areas such as machine learning, fuzzy logic, artificial neural networks are used widely.

The aim of this thesis that need to investigate the Internet of Things, Artificial Intelligence and Cyber Security issues which are among the current technology trends; is to develop an Anomaly-based Intrusion Detection System in which artificial intelligence applications can be used to detect cyber attacks against smart devices in IoT ecosystems in order to ensure security of these devices.

December 2019, 72 pages.

Keywords: IoT, Machine Learning, Security

1. GİRİŞ

İnternet genellikle yazılımları, hizmetleri ve insanları bir araya getiren en geniş ağ olarak tanımlanır. Nesnelerin İnterneti (IoT), günlük nesneleri de bu ağa dahil etmesinden dolayı çok hızlı bir şekilde büyümektedir.

IoT kavramı ortaya çıktığından beri teknoloji alanında önemli rol oynamaktadır. Bilişim, elektronik, makine alanlarını bir araya getiren bu ekosistemle beraber günlük hayatı kolaylaştıran akıllı teknolojiler ortaya çıkmış, insan gücünden tasarruf sağlanmış özellikle endüstriyel alanlarda insan hataları azaltılarak iş verimi ve kalitesinin yükseltilmesi de dahil olmak üzere neredeyse her alanda gelişme sağlanmıştır. Bu teknolojinin sağladığı kolaylıklar kötü niyetli kişilerin dikkatinden kaçmamış, IoT konsepti saldırganların da hedefi haline gelmiştir.

Teknolojinin hızlı evrimi ve bileşenleri arasında artan bağlantı, yeni siber güvenlik sorunlarına neden olmaktadır. Bilgisayar saldırılarındaki bu artan eğilimin üstesinden gelmek ve tehdide yanıt vermek için, endüstri uzmanları ve akademisyenler, yüksek doğruluğu düşük karmaşıklık ve zaman verimliliği ile birleştiren Saldırı Tespit Sistemleri (STS) oluşturmak için bir araya gelmişlerdir. Güncel, henüz bilinmeyen saldırıları tespit etmek için STS'ler yapay zeka algoritmalarından faydalanmaktadır. Hazırlanan tez yapay zeka alanlarından makine öğrenmesi algoritmalarının uygulanması ile ilgilidir.

Bu tezde yer alan "*Genel Kısımlar"* başlığının ilk bölümünde IoT ekosistemiyle ilgili temel kavramlar kısaca açıklandıktan sonra; IoT mimarisine, IoT mimarisiyle ilgili güvenlik konularına değinilmiş, daha sonra IoT protokol yığını, IoT'de güven yönetimi ve IoT için mevcut platformlar açıklanmıştır. Sonraki bölümde Saldırı Tespit Sistemleri (STS) açıklanmış; STS tarihçesinden, STS gereksinimlerinden, STS kararlarından ve STS yanıtlarından bahsedilmiş, STS teknikleri açıklandıktan sonra STS'ler sınıflandırılmıştır. Bölümün sonunda ise STS'lerin avantajları ve dazavantajları açıklanmıştır. *"Malzeme ve Yöntem"* kısmında Makine Öğrenmesi'ne değinilmiş, Makine Öğrenmesi algoritmaları açıklandıktan sonra makine öğrenmesi algoritmalarından bazılarının mevcut veri sistemine uygulanması sonucu elde edilen sonuçlar *"Bulgular"* kısmında açıklanmış *"Tartışma ve Sonuç"* kısmında ise bu sonuçlar değerlendirilmiştir.

2. GENEL KISIMLAR

2.1.NESNELERİN İNTERNETİ (INTERNET OF THINGS-IOT)

Nesnelerin İnterneti çeşitli görevleri gerçekleştirmek amacıyla veri paylaşımı yapan birbirine bağlı nesnelerin ve insanların bir ağı olarak tanımlanabilir. Bu ağın temel amacı çeşitli görevleri gerçekleştirerek günlük yaşam kalitemizi artırmaktır. Ev ürünleri, ulaşım, lojistik hizmetler, sağlık, akıllı şehirler, endüstriyel uygulamalar gibi birçok alanda uygulaması bulunan IoT'nin ulaşabileceği sınırlar neredeyse üretilebilecek tüm nesnelere ulaşabilmektedir. Nesnelerin, internetle buluşarak nesnelerin interneti konseptine dahil olmasının sağlanması için gereken şey sadece hayal gücüdür. IoT ekosistemi için basit bir örnek olarak ulaşım sistemlerini göz önünde bulundurursak, daha güvenli bir ulaşım sistemi sağlamak için ulaşımda yer alan birçok akıllı aracı ve akıllı yönetim sistemlerini görebilmemiz mümkündür.

2.1.1. IoT Ekosisteminde Yer Alan Temel Kavramlar

IoT ekosistemini oluşturan temel kavramlar; nesneler, sensörler, RFID (Radio Frequency Identification), kontrolör, aktivatör, bulut bilişim, sis bilişim ve büyük veridir. Bu bölümde bazı temel kavramlar açıklanacaktır.

2.1.1.1. Nesneler

Nesneler, akıllı kararlar almak için internetle ve birbirleriyle bağlantı kurabilen fiziksel cihazlardır. Bu cihazlar verileri algılar, toplar, içeriğe duyarlı hale gelir ve hem insanlara hem de makinelere yardımcı olmak için çeşitli bilgiler sağlar.

Nesneler genellikle masaüstü bilgisayarlar, dizüstü bilgisayarlar, akıllı telefonlar, tabletler ve bilgisayar kümeleri gibi çeşitli geleneksel bilgisayar ve bilgisayar benzeri cihazlardan oluşmaktadır. IoT, geleneksel olarak internete bağlı olmayan nesneler ve cihazlar dahil olmak üzere tüm nesne türlerini içermektedir. Cisco, fiziksel nesnelerin yüzde 99'unun bir gün internete bağlanacağını tahmin etmektedir [1].

İç sunucular ve dış çevre ile etkileşime geçmek için gömülü teknolojiden yararlanan bu nesneler ağ yeteneğine de sahiptir. Güvenilir ve kullanılabilir bir ağ platformu üzerinden iletişim kurabilirler.

Amaçlanan IoT çözümünün bir parçası olan tüm cihazların iletişim kurabilmeleri için birbirlerine bağlı olmaları gerekmektedir. Kablolu veya kablosuz olmak üzere cihazların bağlantı kurabileceği iki yöntem bulunmaktadır. Geleneksel olarak ağa bağlı olmayan cihazlar için sensörler, RFID'ler ve denetleyiciler gerekmektedir.

2.1.1.2. Sensörler

IoT'de, sensör adı verilen başka tür bir cihaz, veri ağına bağlanmalıdır. Sensörler, fiziksel özellikleri ölçmek ve bu bilgileri elektriksel veya optik sinyallere dönüştürmek için kullanılabilecek nesnelerdir.

Sensörler, bilgisayar harici varlıklardan veri toplamak için kullanılan yöntemlerden biridir. Ortamların fiziksel yönlerini bilgisayarlar tarafından işlenebilen elektrik sinyallerine dönüştürürler. Sensörler için bazı örnekler toprak nemi sensörleri, hava sıcaklığı sensörleri, radyasyon sensörleri ve hareket sensörleri olabilir. IoT kapsamında her türden sensör, geleneksel olarak bağlantısı olmayan cihazların internete bağlanmasında önemli bir rol oynayacaktır.

Sensörler tipik olarak önceden programlanmış özel talimatlarla satın alınır; ancak, bazı sensörler hassasiyet derecelerini veya geri besleme sıklıklarını değiştirecek şekilde yapılandırılabilir. Hassasiyet ayarı, ölçülen miktar değiştiğinde sensör, çıkışının ne kadar değiştiğini gösterir. Sensör ayarlarını lokal olarak veya uzaktan değiştirmek için Grafiksel Kullanıcı Arayüzü (Graphical User Interface-GUI) içeren bir kontrolör kullanılır.

Sensörler, karbon monoksit, karbon dioksit, oksijen, metan, hidrojen, amonyak ve hidrojen sülfür gibi kimyasal seviyeleri tespit edebilir; yakıt kullanımını, motor arızalarını ve motor yükünü; akımı, sıcaklığı ve parlaklığı ölçebilir; toprak nemi, yaprak ıslaklığı, güneş radyasyonu, atmosfer basıncı ve gövde çapını algılayabilir.

2.1.1.3. Radyo Frekansı Tanımlama (RFID)

IoT için birincil etkinleştirme teknolojilerinden biri olan Radyo Frekansı Tanımlama (Radio Frequency Identification-RFID), radyo frekansı ile tanımlamayı kullanan popüler bir sensör türüdür. RFID, küçük kodlanmış etiketler (RFID etiketleri) ve bir RFID okuyucu arasında bilgi iletmek için radyo frekansı elektromanyetik alanları kullanır [1]. Genellikle, RFID etiketleri evcil hayvan takip cihazları gibi içine gömülü oldukları cihazları tanımlamak ve izlemek için kullanılır. Etiketler küçük olduğu için, kıyafet ve nakit para dahil hemen hemen her şeye yapıştırılabilirler. Bazı RFID etiketlerinde pil yoktur. Etiketin bilgi iletmesi için gereken enerji, RFID etiket okuyucusu tarafından gönderilen elektromanyetik sinyallerden elde edilir. Etiket bu sinyali alır ve yanıtı güçlendirmek için aldığı sinyalin enerjisinin bir bölümünü kullanır.

Uygulamaların çoğunda barkodun yerini almakta olan RFID, barkodun aksine, radyo frekansına dayanır; bu nedenle, çalışmak için görüş hattı gerektirmez. IoT ile entegrasyon içinde RFID kullanarak, cihaz güvenliğini artırmak için kimlik doğrulama mekanizmaları eklemek mümkündür [2]. Son birkaç yılda, IoT cihazlarını güvenlik tehditlerine karşı koruyabilen RFID için çeşitli karşılıklı kimlik doğrulama şemaları önerilmiştir [3-5]. IoT, RFID'nin diğer nesneler tarafından tanımlanma, onları tanımlama ve onlarla iletişim kurma yeteneğini dikkate alır. Günümüzde IoT uygulamalarının çoğu, bilgi gönderme, alma ve saklamaya dayanmaktadır [6]. IoT'nin karşılaştığı başlıca geliştirme sorunları ise şunlardır [7,8]:

i. **Ölçeklenebilirlik:** Her geçen gün adresleme, servisler ve veri yönetimi gibi zorlukları artıran daha fazla cihaz IoT ağına katılmaktadır.

ii. **Heterojenlik:** IoT ağları, anlamlı bilgilerle işlenmesi gereken farklı türde verileri toplayan ve depolayan farklı cihaz türlerinden oluşmaktadır.

iii. **Güç ve kaynak kısıtlamaları:** IoT cihazlarında küçük bellek ve düşük güç kaynağı vardır, bu nedenle iletişim sırasında güç kullanımının en aza indirilmesi gerekmektedir.

iv. **RFID sistemindeki güvenlik açıkları:** RFID okuyucusu, etiketler ve sunucu arasında bir araç görevi görür, eğer böyle durumlarda etiket sayısı artarsa sistem

daima ölçeklenebilir olmalıdır. Her etiket talebinde olduğu gibi, sunucunun etiketi tanımlamak için doğrusal bir arama yapması gerekir ve bu arama işleminin süresi etiket sayısı ile artar ve bir doğrulama protokolünün performansını etkileyerek bilişimsel aşırı yüklenmeyi arttırır [9].

Esnekliği ve düşük güç gereksinimleri nedeniyle RFID etiketleri, bilgisayar harici cihazların RFID okuyucu cihazlara bilgi vermelerini sağlayarak bunları bir IoT çözümüne bağlamak için iyi bir seçenektir. Örneğin, otomobil gövdelerine arabaların montaj hattı boyunca daha iyi izlenmesini sağlayacak RFID etiketleri ekleyen otomobil fabrikalarıyla artık yaygın bir şekilde karşılaşılmaktadır [1].

İlk nesil RFID etiketleri "bir kez yaz defalarca oku (write once read many)" şeklindeydi. Bu, bir kez fabrikada programlanabilecekleri, ancak sahada değiştirilemeyecekleri anlamına gelmekteydi. Daha yeni RFID etiketleri, 40 ile 50 yıl sürebilen ve 100.000'den fazla defa yazılabilen tümleşik devrelere sahip, "defalarca yaz defalarca oku (write many read many)" şeklindedir. Bu etiketler, üretim tarihi, konum izleme geçmişi, çoklu hizmet döngüsü ve mülkiyet gibi bağlı oldukları varlığın geçmişini etkin bir şekilde depolayabilir [1].

2.1.1.4. Kontrolör

Sensörler ölçüm yapmak, ölçüm sonucunda elde ettikleri verileri sinyale çevirmek ve ardından sinyalleri de kontrolör adı verilen ana cihaza göndermek için programlanabilir. Kontrolör, sensörlerden veri toplamaktan ve internet bağlantısı sağlamaktan sorumludur. Kontrolörler anında karar verme yeteneğine sahip olabilir veya analiz için daha güçlü bir bilgisayara veri gönderebilirler. Bu daha güçlü bilgisayar, kontrol cihazı ile aynı yerel ağda olabilir veya yalnızca bir internet bağlantısı üzerinden erişilebilir olabilir.

İnternete ve ardından veri merkezindeki daha güçlü bilgisayarlara ulaşmak için, kontrol cihazı önce yerel bir yönlendiriciye veri gönderir. Bu yönlendirici, yerel ağ ve internet arasında arayüz oluşturur ve aralarında veri iletebilmelerini sağlar.

2.1.1.5. Aktivatör

IoT ekosisteminde yer alan bir başka cihaz da aktivatördür. Bir aktivatör, belirli bir talimat setine dayanarak bir mekanizmayı veya sistemi hareket ettirmek veya kontrol etmek için

kullanılabilen temel bir motordur. Aktivatörler “işlerin gerçekleşmesi için” fiziksel bir işlev gerçekleştirebilir.

IoT'de kullanılan üç tip aktivatör vardır:

i. **Hidrolik-** Mekanik hareketin gerçekleşmesi için akışkan basıncının kullanıldığı aktivatör türüdür.
ii. **Pnömatik**- Mekanik çalışmanın sağlanması için yüksek basınçta basınçlı havanın kullanıldığı aktivatör türüdür.
iii. **Elektriksel**- Elektrik enerjisini mekanik enerjiye dönüştüren bir motorla çalışan aktivatör türüdür.

Aktivatörün temel işlevi bir sinyal almak ve bu sinyale göre belirlenmiş bir eylem gerçekleştirmektir. Aktivatörler genellikle veri işleyemez; gerçekleştirdikleri eylemin sonucu, alınan bir sinyale dayanmaktadır. Aktivatör tarafından gerçekleştirilen harekete tipik olarak kontrolörden gelen bir sinyal neden olur.

2.1.1.6. Bulut Bilişim

Bulut bilişim, sınırsız alan sağlayarak isteğe bağlı servis ve ortak kaynaklar havuzu ile yaygın ağları etkinleştiren bir teknolojidir. Bu teknolojinin IoT ile entegrasyonu, birçok kısıtlamaya sahip olan IoT mimarisinin gereksinimlerini karşılar. Bulut bilişim, IoT uygulamaları, servisleri ve kaynak yönetimi için çok etkili bir çözüm sunar [10-13]. IoT ve Bulut entegrasyonu ile ilgili konuların bazıları şunlardır [14-17]:

i. **Standardizasyon:** Nesnelerin çoğu, mevcut senaryoda, depolama sorununu çözen ve çeşitli hizmetler sunan buluta bağlanır. Bu nedenle, heterojen platformlar arasında birlikte çalışabilirliği sağlamak ve hizmetleri geliştirmek üzere akıllı nesneler arasında ara bağlantıyı kolaylaştırmak için standart protokollere, arayüzlere ve API'lere ihtiyaç duyulmaktadır.
ii. **Entegrasyon:** Gerçek zamanlı veri üreten ve işleyen bulut ile akıllı nesnelerin birbirine bağlanması, bilişsel yetenekleri ve akıllı karar vermeyi gerektirir. Bu nedenle, standart iş akışlarıyla ortak bir entegrasyon yönteminin olması gerekir. Bulut ve IoT'nin birleşimi, bütünleşik hizmetleri farklı varlıklar arasında dağıtmak

ve entegre senaryoda işlem ve verileri yönetmek için genel bir platforma ihtiyaç duyar.

iii. **Ölçeklenebilirlik:** Akıllı cihazlar çok miktarda veri toplar ve bu toplanan verileri ilgili uygulama veya olayla ilişkilendirmek için verimli teknikler gerektirir. Bulut bilişim, akıllı cihazlara ve son kullanıcılara göre ölçeklenebilirliğin korunmasına yardımcı olan genişletilebilir destek ve veri yönetimi sağlar.

iv. **Güç ve Enerji Kısıtlamaları:** IoT uygulamaları hemen hemen her zaman veri iletimi ve iletişim içerir, böylece güç tüketimi ve enerji gereksinimleri artar. Bulut bilişim, güç tüketimini azaltmak ve önbelleğe alınmış verilerin yeniden kullanılmasını sağlamak için veri sıkıştırma, verimli iletişim ve önbellekleme mekanizmaları gibi çeşitli hizmetler sunar. Bulut ayrıca uzun süreli veri izlemenin üstesinden gelmek için ara katman teknolojilerini de kullanır.

v. **Güvenlik ve Mahremiyet:** Güvenlik ve mahremiyet, herhangi bir uygulamada her zaman birincil öneme sahiptir. IoT ve Bulut Bilişim'in entegrasyonu, güvenlik tehditlerini tamamen daha ciddi hale getiren gerçek dünyadaki verileri ve olayları birleştirmektedir. Bulut Bilişim, uygun şekilde tasarlanmış politikalar ve yetkilendirilmiş roller sağlar; böylece yalnızca meşru kullanıcılar hassas verilere erişebilir.

2.1.1.7. Sis Bilişim

Bulut bilişim, geleneksel istemci-sunucu modelinin birçok sorununu çözmüş olmakla birlikte; acil, yerel bir yanıt gerektiren gecikmeye duyarlı uygulamalar için en iyi seçenek olmayabilir.

IoT ekosistemi, konum farkındalığı ve asgari gecikmeye ek olarak, mobilite desteği ve coğrafi dağılım gerektirir. IoT'deki cihazlar gerçek zamanlı veri ve servis kalitesi mekanizmasına ihtiyaç duymaktadır. IoT, neredeyse her şeyi izleyebilen veya ölçebilen çok fazla sayıda IP-etkin aygıt içerir. Bununla birlikte, bu cihazların ortak noktalarından biri, tüm dünyaya dağıtılmış olmalarıdır.

Buradaki en önemli zorluklardan biri, bu cihazlar ile verinin analiz edilebileceği veri merkezleri arasında bağlantılar oluşturmaktır. Bu cihazlar çok miktarda veri üretebilir. Örneğin, bir jet motoru; sadece 30 dakikada, performansı ve durumu hakkında 10 terabayt veri üretebilir [1]. IoT cihazlarından gelen tüm verilerin analiz edilmek üzere Bulut'a

aktarılması ve ardından kararların buluttan alınıp uygulanması verimsiz olacaktır. Bunun yerine, analiz çalışmasının bir kısmı, örneğin sahada çalışmak üzere inşa edilmiş endüstriyel güçte yönlendiriciler üzerinde, en başta gerçekleştirilmelidir.

Sis bilişim, hızlı bir yanıt gerektiren daha kolay görevler gerçekleştiren ağ kenarına daha yakın bir dağıtılmış bilişim altyapısı oluşturur. Ağlardaki veri yükünü azaltır. Ağ bağlantıları kaybolduğunda IoT cihazlarının çalışmasına izin vererek esnekliği artırır. Ayrıca, hassas verilerin ihtiyaç duyulan kenarın ötesine taşınmasını önleyerek güvenliği artırır.

Akıllı trafik ışığı sistemini, sis bilişimin iyi bir kullanımı olarak düşünebiliriz. Akıllı trafik ışığı sistemi, gerçek zamanlı etkileşimler için destek gösterir. Sistem yerel olarak çeşitli sensörlerle etkileşime girer. Sensörler, yayaların ve bisikletlilerin varlığını algılar; yaklaşmakta olan araçların mesafesini ve hızını ölçer. Sistem ayrıca trafik ışıklarıyla koordine etmek için komşu ışıklarla etkileşime girer. Bu bilgilere dayanarak, akıllı ışık yaklaşan araçlara uyarı sinyalleri gönderir ve kazaları önlemek için kendi döngüsünü değiştirir. Sis'teki komşu akıllı trafik ışığı sistemleriyle yeniden koordinasyon yapmak, döngünün herhangi bir şekilde değiştirilmesine izin verir. Akıllı trafik ışığı sistemi tarafından toplanan veriler, gerçek zamanlı analiz yapmak için yerel olarak işlenir. Bu sistemlerin kümelerinden elde edilen veriler, uzun vadeli trafik düzenlerini analiz etmek için Bulut'a gönderilir [1].

2.1.1.8. Büyük Veri

IoT tarafından üretilen büyük veriler, toplanan verilerdeki heterojenlik ve gürültü nedeniyle normal büyük verilerden farklı niteliktedir [2]. 2030 yılında konuşlandırılan toplam sensör sayısının IoT'yi büyük verilere büyük bir iştirakçi yapacak şekilde trilyonlarca olacağı HP tarafından tahmin edilmektedir [18,19]. IoT verisini büyük verilere eşleyen temel özellikler şunlardır [20]:

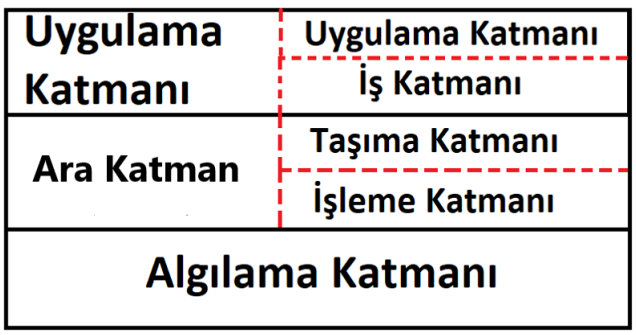
- Devasa miktarda veri toplayan çok sayıda sensör.
- Toplanan verilerin çoğunun yapılandırılmamış ve gereksiz olması.
- Verilerin ancak analizden sonra faydalı bilgilere dönüştürülebilmesi.

IoT potansiyel olarak akıllı şehirler, sanayi, tarım, ulaşım, sağlık, perakende vb. alanlarda büyük veri kaynaklarından biridir. IoT tarafından üretilen büyük verilerin sergilediği ana özellikler ise şunlardır [21–23]:

i. **Büyük ölçekli veri:** IoT ağlarında üretilen veriler, büyük miktarlarda heterojen veri toplayan çok fazla veri toplama aracı gerektirdiğinden büyük ölçekli olduğu düşünülmektedir.
ii. **Heterojenlik:** IoT cihazları tarafından toplanan veriler, farklı özelliklere sahip verileri toplayabilen ve analiz edebilen bir dizi veri toplama cihazı gerektiren herhangi bir format ve büyüklükteki metin, ses, video vb. olabilir.
iii. **Zaman ve mekan korelasyonu:** Zaman ve mekan istatistiksel analiz için; zaman ve mekan arasındaki ilişki de veri analizi için önemli bir yere sahip olduğundan, konum denetleyicileri olan tüm veri toplama cihazlarının ve tüm veri paketlerinin zaman damgaları vardır.
iv. **Daha az miktarda etkili veri:** Cihazlar tarafından elde edilen çok miktarda verinin yalnızca belirli bir miktarı gerçekten yararlıdır. Örneğin, bir trafik izleme videosu elde edilirken sadece kural ihlallerini veya kazaları gösteren kareler diğerlerine göre daha önemlidir [24].

2.1.2. IoT Mimarisi

IoT için standart bir mimari olmamakla birlikte; üç, dört veya beş katmandan oluşan kesin olmayan birkaç mimari vardır. Başlangıçta, en popüler IoT mimarisi Sensör Katmanı, Ara Katman ve Uygulama Katmanı olmak üzere üç katmandan oluşmaktaydı [2]. Şekil 2.1'de üç katmanlı ve beş katmanlı mimariler bir arada yer almaktadır.



Şekil 2.1: IoT mimarisi.

i. **Algılama/Sensör Katmanı:** Bu katman sensörler veya RFID etiketlerinden oluşur. Katmanın amacı, kendilerinden bilgi toplanacak nesneleri benzersiz şekilde tanımlamaktır. Ayrıca, ortamdan diğer verileri gönderip alırlar ve işlem için üst katmanlara iletirler [25].
ii. **Ara Katman/Ağ Katmanı:** Bu katmanın amacı IoT için ağ desteği ve protokol yığını sağlamaktır. Diğer mimarilerde; sensör katmanı tarafından toplanan bilgilerle ilgilenen işleme katmanı ile Bluetooth ve Wi-Fi gibi teknolojileri dağıtan taşıma katmanı olmak üzere iki kısma ayrılabilir. Algılama katmanına veri gönderir ve alır. Ayrıca sistemdeki nesnelere adres atamak için IPv6 adreslemeyi kullanır [25,26].
iii. **Uygulama Katmanı:** Bu katmanın amacı akıllı şehirler, sağlık gibi uygulamaya özgü işlevleri yerine getirmektir. Bazı mimarilerde bu katman uygulama yönetimi yapan; güvenlik ve gizlilik ile ilgilenen iş katmanı ve farklı uygulamalar arasını ayırt eden uygulama katmanı olarak iki kısım halinde gerçekleştirilir [25,27].

IoT mimarisinin her katmanında, IoT uygulamalarının güvenliğinin sağlanması için ele alınması gereken güvenlik ve gizlilikle ilgili bazı sorunlar bulunmaktadır. Tüm bu sorunlar, dikkate alınarak sistem tasarımının ilk aşamasında düzeltilmelidir. Mevcut IoT mimarisi, bir IoT ağı için başlangıçta ve düzenli aralıklarla uygun güvenlik kontrollerine ihtiyaç duymaktadır [28].

Algılama katmanlarında kullanılan birincil teknolojiler WSN, RFID gibi algılama ve tanımlama teknikleridir. Bu katmanın karşılaştığı en yaygın tehdit türleri ise şunlardır [28]:

i. **Düğüm Yakalama:** Ağda mevcut olan cihazların riske girme olasılıkları yüksektir ve bu da tüm ağın güvenliğini tehlikeye sokarak önemli bilgilerin sızmasına neden olabilir.
ii. **Sahte Düğüm ve Kötü Amaçlı Veriler:** Saldırgan; mevcut sisteme, kötü amaçlı kodları ve bilgileri ağ üzerinden dolaştırarak tüm sisteme bulaştırabilecek kötü amaçlı bir cihaz ekleyebilir.
iii. **Servis Dışı Bırakma Saldırıları (DoS-DDoS):** DoS ve DDoS saldırıları, ağ üzerinden yapılan en yaygın ve en zararlı saldırılardır. Bu saldırılar ağ kaynaklarının tükenmesine ve hizmetlerin kullanılamamasına yol açmaktadır.
iv. **Tekrarlama Saldırıları:** Saldırgan, ağ güvenliğini ve kimlik doğrulama sistemlerini tehlikeye atmak için hedef düğüme bir önceki mesajı tekrar tekrar gönderebilir.

Ara katmanda; gizlilik tehditleri, bütünlük ve kullanılabilirlik ele alınmalıdır. Gizlice dinleme, ortadaki adam, DoS/DDoS, ağa sızma gibi saldırılar bu katmandaki ortak tehditlerdir [26,28,29]. Bu katmanda ele alınması gereken başlıca sorunlar ise şunlardır:

i. **Heterojenlik:** Farklı teknolojilerin ve protokollerin kullanımı nedeniyle güvenlik ve ağ koordinasyonunu korumak zordur. Bu da sistemi savunmasız hale getirir.
ii. **Ölçeklenebilirlik Sorunları:** IoT, çok fazla sayıda cihazdan oluşur ve çok fazla cihaz farklı zamanlarda ağa girebilir veya ağdan ayrılabilir; bu da kimlik doğrulama eksikliği, ağ tıkanıklığı gibi sorunları ortaya çıkarır.
iii. **Veri İfşası:** Sosyal mühendislik tekniklerini kullanarak, saldırgan, ağdan hassas bilgiler elde edebilir. Bu cihazlar topluca büyük miktarda veriye sahip olduklarından, belirli veri alma teknikleri kullanılarak cihazlardan bilgi çıkarılabilir.

Uygulama katmanı, uygulamanın gerekliliklerine göre farklı güvenlik standartlarına ihtiyaç duyar ve bu da uygulamanın zor ve karmaşık olmasına sebep olur. Bu katmandaki güvenlik ve gizlilik sorunlarından bazıları şunlardır [28,29]:

i. **Karşılıklı kimlik doğrulama ve cihaz tanımlama:** Her uygulamanın, farklı derecelerde erişim ayrıcalıkları gerektiren farklı bir kullanıcı kümesi vardır. Bu nedenle, istenmeyen erişimleri engellemek için etkili kimlik doğrulama planları uygulanmalıdır.
ii. **Bilgi gizliliği:** Her iletişim için kullanıcı gizliliği sağlanmalıdır. Bazen verileri işlemek için kullanılan teknikler saldırıya açık olabilir; bu da veri kaybına neden olur ve uzun vadede sisteme büyük zararlar verebilir.
iii. **Veri yönetimi:** Büyük veri toplamalarından dolayı, veri yönetimi için çok fazla kaynak ve karmaşık algoritmalar gerekeceğinden sistem karmaşıklığı artar ve veri kayıpları ortaya çıkabilir.
iv. **Uygulamaya özel güvenlik açıkları:** Bir uygulama için modüller geliştirirken, kullanıcı tarafından bilinmeyen bazı güvenlik açıkları bırakılabilmektedir. Bunların daha sonra saldırganlar tarafından tespit edilerek kullanılması mümkündür.

2.1.3. IoT İçin Protokol Yığını

İki cihaz bir ağ üzerinden iletişim kurduğunda, ilk olarak önceden tanımlanmış kurallar veya protokoller kümesi üzerinde anlaşmalıdırlar. Protokoller, cihazların kullandığı diyaloğun özelliklerine özgü iletişim kurallarını ifade eder. Günlük kişisel iletişimlerde, telefon görüşmesi gibi bir ortam üzerinden iletişim kurmak için kullanılan kuralların, mektup göndermek gibi başka bir ortamın kullanılmasıyla ilgili protokollerle aynı olması gerekmez. Protokoller, mesajların nasıl iletilip alındığı ile ilgili detayları tanımlar. İnsanların iletişim kurmak için dili nasıl kullandıklarına benzer olarak, protokoller aygıtların nasıl iletişim kurduğuna ilişkin kuralları içerir. Bir iletişim işlevini gerçekleştirmek için gereken; birbiriyle ilişkili protokol grubuna protokol kümesi adı verilir. Protokol kümeleri, ağ aygıtları arasında birlikte çalışabilirliği sağlamaya yardımcı olur. Bir protokol kümesi içindeki bireysel protokoller, üreticiye özel ve tescilli olabilir. Tescilli kavramı, bu bağlamda, bir şirketin veya satıcının, protokolün tanımını ve nasıl çalıştığını kontrol ettiği anlamına gelir. Bazı özel protokoller, sahibinden izin alınarak farklı kuruluşlar tarafından kullanılabilir. Diğerleri yalnızca tescilli satıcı tarafından üretilen ekipmanlara uygulanabilir [1].

Bugün IoT'de kullanılabilen birkaç kablosuz ağ protokolü vardır. Bu protokollerin özellikleri büyük ölçüde değişir. Tablo 2.1'de yer alan IoT protokol yığını aşağıda açıklanmaktadır [30].

Tablo 2.1: IoT protokol yığını.

<table>
<tr><td colspan="2">Uygulama Protokolleri</td><td>DDS</td><td>CoAP</td><td>AMQP</td><td>MQTT</td><td>MQTT-SN</td><td>XMPP</td><td>HTTP REST</td></tr>
<tr><td colspan="2">Hizmet Bulma Protokolleri</td><td colspan="4">mDNS</td><td colspan="3">DNS-SD</td></tr>
<tr><td rowspan="4">Altyapı Protokolleri</td><td>Yönlendirme Protokolleri</td><td colspan="7">RPL</td></tr>
<tr><td>Ağ Katmanı</td><td colspan="4">6LoWPAN</td><td colspan="3">IPv4/IPv6</td></tr>
<tr><td>Bağlantı Katmanı</td><td colspan="7">IEEE 802.15.4</td></tr>
<tr><td>Fiziksel Katman</td><td colspan="2">LTE-A</td><td>EPCglobal</td><td>IEEE 802.15.4</td><td colspan="2">Z-Wave</td></tr>
<tr><td colspan="2">Etkili Protokoller</td><td colspan="5">IPsec</td><td>IEEE 1905.1</td></tr>
</table>

2.1.3.1.Uygulama Protokolleri

Genişletilebilir Mesajlaşma ve Varlık Protokolü (Extensible Messaging and Presence Protocol-XMPP): Telekom yeniliklerinde önemli bir unsur olan XMPP, çoğu anlık mesaj uygulaması için tercih edilmektedir. Çok katılımcılı sohbet, sesli ve görüntülü arama ve telekonferans için kullanılan bir IETF anlık mesajlaşma standardıdır. Daha önce Jabber olarak adlandırılan XMPP, bireysel işletim sistemlerinden bağımsız olarak internet üzerinden anında mesajlaşmayı mümkün kılmaktadır [31].

Kısıtlı Uygulama Protokolü (Constrained Application Protocol-CoAP): Bu web aktarım protokolü, HTTP işlevselliklerinin üstünde temsili durum aktarımına (Representational State Transfer- REST) dayanmaktadır. REST, istemciler ve sunucular arasında veri alışverişini HTTP üzerinden kolaylaştırır. Mobil ve sosyal ağ uygulamalarında kullanılan REST önbelleklenebilir bağlantı protokolü, durumsuz istemci-sunucu mimarisine dayanır. Ayrıca, HTTP GET, POST, PUT ve DELETE yöntemlerini kullanarak belirsizliği ortadan kaldırır [32].

Mesaj Kuyruklama Telemetri İletimi (Message Queuing Telemetry Transport- MQTT): Bu mesajlaşma protokolü, gömülü cihazları ve ağları uygulamalar ve ara yazılımlarla birleştirir. Özellikle sensör ağları için tanımlanmış olan MQTT, IoT ve M2M için bağlantı protokollerini optimize eden yönlendirme mekanizmaları (bire bir, bire çok, çoktan çoğa) sunar. Yayınlama/abone olma modeli, geçişi ve uygulamayı kolaylaştırır ve güvenilmez veya düşük bant genişliği bağlantıları kullanan kaynak kısıtlı cihazlar için uygundur [33].

Gelişmiş Mesaj Kuyruklama Protokolü (Advanced Message Queuing Protocol-AMQP): Bu açık standart uygulama katmanı protokolü mesaj odaklı ortamlara odaklanmaktadır. Mesaj alışverişi için TCP gibi güvenilir bir taşıma protokolü gerektirir. Bir kablo seviyesi protokolü tanımlayarak, AMQP uygulamaları birbirleriyle birlikte çalışabilir [34].

Veri Dağıtım Servisi (Data Distribution Service-DDS): Bu yayınlama/abone olma protokolü, gerçek zamanlı M2M iletişimine izin verir. MQTT veya AMQP gibi diğer protokollerin aksine, DDS, aracı kurumsuz bir mimariye sahiptir ve çok noktaya yayınlara dayanır. Böylece, mükemmel hizmet kalitesi (QoS) ve yüksek güvenilirlik sunar. Bu uygulama katmanı protokolleri, belirli senaryolar ve uygulamalar için kendilerine uygun özelliklere sahiptir [35].

2.1.3.2.Hizmet Bulma Protokolleri (Service Discovery Protocols-SDP)

Servis bulma protokolleri, IoT cihazları üzerinden mevcut olan kaynakları ve hizmetleri kaydeder ve keşfeder. Baskın protokoller multicast DNS (mDNS) ve DNS Hizmet Bulma (DNS-SD)'dir. Her ne kadar kaynak bakımından zengin cihazlar için tasarlanmış olsa da bu iki protokol IoT ortamları için uyarlanabilir.

Multicast DNS (mDNS): Sohbet gibi bazı IoT uygulamaları için mDNS tek noktaya yayın yapan DNS sunucusunun görevini gerçekleştirebilir. DNS alanı yerel olarak kullanıldığından, mDNS ekstra harcama veya yapılandırma olmadan esneklik sunar [36]. Protokol, yerleşik internet tabanlı cihazlar için çok uygundur, çünkü:

- Aygıtları yönetmek için el ile yapılanmaya ya da ek yönetime gerek yoktur.
- Altyapı olmadan çalışır.
- Altyapı arızası olursa çalışmaya devam eder.

DNS Service Discovery (DNS-SD): DNS tabanlı servis bulma (DNS-SD), mDNS kullanan müşterilerin aradıkları hizmetleri bulmalarına yardımcı olur. Bu protokolü kullanarak, istemciler belirli bir ağda ihtiyaç duydukları hizmetleri keşfetmek için standart DNS mesajlarını kullanabilirler. DNS SD, mDNS gibi, harici yönetim veya yapılandırma olmadan makineleri bağlayabilir [37].

2.1.3.3.Altyapı Protokolleri

Düşük Güç ve Kayıplı Ağlar Üzerinde Yönlendirme Protokolü (Routing Protocol for Low-Power and Lossy Networks-RPL): RPL, kayıplı linkler üzerinden kaynak kısıtlı düğümler için IPv6'ya dayanan, linkten bağımsız bir yönlendirme protokolüdür. Sağlam bir topolojiye dayanan protokol, çok noktadan noktaya, noktadan çok noktaya ve noktadan noktaya gibi basit ve karmaşık trafik modellerini destekler [38].

6LoWPAN: Bu protokol, IPv6'nın Düşük Güçlü WPAN'lar üzerinden gerektirdiği eşleme hizmetlerini belirtir. IPv6 paketlerini IEEE 802.15.4 teknik özelliklerine uyarlayan bir adaptasyon katmanıdır. Standart, iletim ek yükünü azaltmak için başlık sıkıştırması, IPv6 maksimum iletim birimi gereksinimini karşılamak için parçalanma ve çoklu atlama iletimini desteklemek için bağlantı katmanına iletme sağlar [39].

IEEE 802.15.4: Bu protokol, Ortam Erişim Kontrolü (Medium Access Control -MAC) için bir alt katman ve düşük hızlı kablosuz kişisel alan ağları (low-rate wireless personal area networks-LR-WPAN) için bir fiziksel katman belirlemek için oluşturulmuştur [40]. IEEE 802.15.4 birçok düğümü yönetebilir ve farklı platformlarda güvenilir iletişim ve çalıştırılabilirlik sağlar. IoT, M2M teknolojisi ve kablosuz sensör ağları (WSN'ler) düşük güç tüketimi, düşük veri hızı, düşük maliyet ve yüksek mesaj verimi nedeniyle bu protokolü tercih eder. IEEE 802.15.4 yüksek düzeyde güvenlik, şifreleme ve kimlik doğrulama hizmetleri sunarken, QoS garantisi vermez.

Bluetooth Düşük Enerji (Bluetooth Low Energy-BLE): Bluetooth Smart olarak da adlandırılan BLE, daha uzun süre çalışabilmesi için minimum güçle kısa mesafeli bir radyo kullanmaktadır. Bu verimlilik, önceki sürümlerine kıyasla IoT uygulamaları için iyi bir seçenek sağlar. BLE'nin kapsama alanı, yaklaşık 100 metre menzil ve 15 kat daha kısa gecikme süresi ile klasik Bluetooth teknolojisinin kapsama alanının on katıdır. Akıllı telefon üreticileri, çoğu akıllı telefon modelinde mevcut olan BLE standardını hızla geliştirmiştir [41].

EPCglobal: Elektronik Ürün Kodu (Electronic Product Code-EPC), tedarik zinciri yönetimindeki maddeleri tanımlamak için kullanılan bir RFID etiketinde depolanan benzersiz bir kimlik numarasıdır. Temel mimari, RFID etiketleri ve ürün bilgilerini paylaşmak için okuyucular içeren internet tabanlı RFID teknolojilerine dayanır. Nesne kimliklerini ve hizmet keşfini desteklemenin yanı sıra, EPCglobal'in açıklığı, ölçeklenebilirliği, birlikte çalışabilirliği ve güvenilirliği, bu protokolü IoT'nin geleceği için umut verici bir teknik haline getirmektedir [42].

Uzun Dönem Evrim-Gelişmiş (Long Term Evolution-Advanced - LTE-A) : Bu hücresel iletişim protokolleri kümesi, makine tipi iletişim ve IoT altyapıları için uygundur. Hizmet maliyeti ve ölçeklenebilirlikteki diğer hücresel çözümlerden daha iyi performans gösterir. LTE-A, uzun vadeli altyapı dayanıklılığının beklendiği akıllı şehir uygulamaları için özellikle uygundur [43]. LTE Kategori M ve NarrowBand IoT (NB-IoT) teknolojilerindeki son gelişmeler, daha uzun pil ömrü, daha yüksek hız ve kablosuz iletişimde kalite için yeni seçenekler sunar [44].

Z-Wave: Ev otomasyon ağları için olan bu düşük güçlü kablosuz iletişim protokolü, daha küçük ticari alanlarda ve akıllı evlerde uzaktan kumanda uygulamasında yaygın olarak kullanılmaktadır. Z-Wave, noktadan noktaya yaklaşık 30 metreyi kapsar ve küçük veri iletimi gerektiren uygulamalar için kullanılır. Bu uygulamalar arasında ışık kontrolü, ev aletleri kontrolü, akıllı enerji ve HVAC (Heating Ventilating and Air Conditioning), erişim kontrolü, giyilebilir sağlık kontrolü ve yangın algılama yer alır [30].

2.1.3.4.Etkili Protokoller (Influential Protocols)

IEEE 1905.1: IEEE 1905.1, yakınsak dijital ev ağları ve heterojen teknolojiler için bir standart olarak tasarlanmıştır. Altta yatan katmanlarda değişiklik yapmak zorunda kalmadan IEEE 1905.1, Ortam Erişim Kontrolü (MAC) topolojilerinin çeşitliliğini gizleyen bir soyutlama katmanı sağlar. Protokol aynı zamanda ortak ev ağı teknolojilerine bir arayüz sağlar ve veri bağlantısının ve fiziksel katman protokollerinin bir arada bulunmasına izin verir [45].

2.1.3.5.Protokoller İçin Güvenlik Gereksinimleri

IoT ağlarının güvenliği, harici güvenlik modelleri uygulamak yerine, her katmanda kullanılan iletişim protokolleri ile ele alınmalıdır. Birincil güvenlik tasarımı gereklilikleri gizlilik, bütünlük, kimlik doğrulama ve anonimlik sağlamaktır. IEEE 802.15.4'teki [46,47] modlar, erişim kontrol fonksiyonları ve zaman senkronizasyonu sayesinde güvenlik sağlanır.

6LoWPAN katmanı için şu ana kadar tanımlanmış hiçbir güvenlik mekanizması yoktur [2]. Bununla birlikte, ilgili RFC'ler güvenlik zorluklarından ve ağ katmanında güvenlik kullanımından bahseder. Örneğin, RFC 6606, güvenlik tasarımlarının zaman senkronizasyonu ve yerelleştirme açısından önemini tartışmaktadır.

RPL protokolü güvenlik modlarını birleştirir. Kontrol mesajı 4 byte güvenlik alanı içerir. Alan kodu, verilen mesaj için güvenlik uygulanıp uygulanmayacağını belirtmek için daha yüksek bir bit değerine sahiptir [49].

CoAP, uygulama katmanında DTLS (Datagram Transport Layer Security) ile bütünleşik güvenliği destekler. DTLS, uygulama düzeyinde gizlilik, bütünlük ve kimlik doğrulama sağlar.

2.1.4. IoT'de Güven Yönetimi

Güvenilir bir IoT ağı sağlamak için aşağıdaki hedeflere ulaşılmalıdır:

a. **Korelasyon ve anlaşma:** Güven yönetimi, IoT ağ düğümleri arasındaki korelasyon ve anlaşma derecesinin değerlendirilmesini sağlar. Aynı zamanda fikir birliğine varmalarına ve birbirleriyle iş birliği yapmalarına yardımcı olur. Güven korelasyonu değerlendirmesi, IoT ağında bulunan tüm kurumları tüm katmanlarda dikkate alır; akıllı ve otonom güven yönetimi için temel bir rol oynar [48,49].
b. **Bilgi edinme:** Güven yönetim sistemi, ağdaki düğümler arasında güvenilir veri aktarımı sağlamalıdır. Bunu başarmak için sistemin hassas ve azimli olmasının yanı sıra verimli veri toplanmasını da sağlaması gerekir. Bu amaca genellikle algılama katmanında ulaşılır [50].
c. **Gizliliğin Korunması:** Güven yönetim sistemi, IoT ağında bulunan hassas verilerin gizliliğinin ve mahremliğinin korunması için politikalara uygun hareket etmelidir [51].
d. **Güvenilir iletişim:** Güven yönetim sistemi, ağdan geçen veri akışının ve mesajların herhangi bir tehdide karşı güvenli olmasını sağlamalıdır. Bu amaçla, ağda herhangi bir kötü amaçlı erişimi kontrol etmek için yetkilendirme sertifikaları ve anahtar yönetimi dağıtılır. Bu güven yönetimi hedefi, IoT'nin gelişimi için temel güvenlik gereksinimlerinden biridir [52].
e. **Hizmet kalitesi:** Güven yönetim sisteminin bu amacı esas olarak uygulama katmanına özeldir, ancak diğer katmanlardan da destek gerektirir. Hizmet kalitesi, hizmetlerin doğru son kullanıcıya doğru zamanda verilmesini sağlar [53].

Diğer tüm ağlarda olduğu gibi, bir IoT ağını korumak için de aşağıda yer alan genel güvenlik gereksinimlerine ihtiyaç duyulmaktadır [54]:

i. **Doğrulama:** Kısıtlı bir kaynağa erişime izin vermeden veya önemli bilgileri açığa çıkarmadan önce; algılama cihazlarının, kullanıcıların ve ağ geçidi düğümlerinin doğrulanmasını içerir.
ii. **Bütünlük:** Bütünlüğün sağlanabilmesi için mesajın veya değerlendirilen varlığın değiştirilmemiş olması gerekmektedir.
iii. **Gizlilik:** Kablosuz iletişim kanalının gizliliği veya mahremiyeti, bilginin izinsiz ifşa edilmemesini gerektirir.

iv. **Erişilebilirlik:** DOS saldırıları altındayken bile, yetkili kullanıcılara ilgili ağ servisleri sağlanabilmelidir.

v. **İnkar edilemezlik:** Zararlı bir varlığın, eylemlerini gizlemesini önlemek amaçlanır.

vi. **Yetkilendirme:** Yalnızca meşru IoT algılama cihazlarının ağ servislerine bilgi sağlayabilmesini ifade eder.

vii. **Tazelik:** Bilgilerin yeni olduğunu ve eski mesajların herhangi bir saldırgan tarafından tekrarlanamadığı anlamına gelir.

Yukarıdaki güvenlik gereksinimleri dışında, aşağıdaki iki önemli güvenlik özelliği de yerine getirilmelidir:

- **İleri gizlilik:** Bir IoT algılama düğümü ağdan ayrıldığı takdirde, çıkışından sonra bu düğümden gelecek mesajların engellenmesi gerekmektedir.
- **Geri gizlilik:** Ağa eklenen yeni bir IoT algılama düğümü, daha önce iletilen herhangi bir mesajı okuyamamalıdır.

Ek olarak, yukarıda belirtilen güvenlik gereksinimlerine dayanarak, IoT ortamı için güvenlik protokollerinin tasarlanmasında çeşitli saldırılar önlenmelidir. Bu saldırıların bazıları şunlardır [54]:

- **Tekrarlama saldırıları (Replay attack):** Tekrarlama saldırıları, saldırganların, iletim sırasında bilgiyi yeniden kullanarak başka bir yetkili varlığı yanlış yönlendirmeye çalıştığı saldırılardır. Bu saldırılar yetkisiz bir üçüncü tarafın iletilen mesajları kaydetme girişimidir.
- **Ortadaki adam saldırıları (Man-in-the-middle attack):** Bu tarz saldırılarda saldırgan, iletilen mesajları durdurur ve mesajların içeriğini değiştirmeye/silmeye çalışır.
- **Aynı giriş kimliği ile birçok kullanıcının giriş yapması (Many logged-in users with the same login-id attack):** Kullanıcı giriş bilgilerini kontrol etmek için doğrulayıcı/parola tablosunu kullanan sistemler, aynı giriş kimliği saldırısı ile giriş yapmış birçok kullanıcıya maruz kalabilir. Bu nedenle, meşru bir kullanıcı sistemdeki diğer meşru kullanıcılar tarafından da kullanılabilecek bir parola tabanlı şemada, aynı giriş kimliğini ve şifreyi kullansa bile, diğer kullanıcılar bu saldırıyı başlatamamalıdır.

- **Çalıntı/kayıp akıllı kart saldırısı:** Eğer saldırgan kaybedilen/çalınan bir akıllı karta sahipse, güç analizi saldırıları gibi teknikleri kullanarak kart hafızasına kaydedilen tüm bilgileri çıkarabilir [55]. Alınan bilgiyi kullanarak, gizli kimlik bilgilerini elde edebilir. Bu nedenle, IoT'deki bir güvenlik protokolünün, akıllı kart çalınsa/kaybolsa bile, saldırganın gizli bilgileri hesaplayamayacağı şekilde tasarlanması önemlidir.
- **Parola tahmin saldırısı:** Parola tabanlı bir şemada, saldırgan meşru olarak kayıtlı bir kullanıcının parolasını, çevrimiçi veya çevrimdışı modda, gizli mesajları ve ayrıca sistemde veya bir kullanıcının akıllı cihazında saklanan kimlik bilgilerini kullanarak tahmin etmeye çalışabilir. Bu nedenle, kimlik bilgileri, kullanıcının akıllı cihazı kaybolsa veya çalınsa bile, saldırgan tarafından kullanıcının gizli kimlik bilgileri alınamayacak şekilde depolanmalıdır.
- **Parola değiştirme saldırısı:** Bu saldırılarda, saldırgan yetkili bir kullanıcının parolasını değiştirmeyi deneyebilir. Örneğin, meşru bir kullanıcının akıllı cihazının, akıllı cihaz tabanlı bir şeması ihlal edilirse, saldırgan kullanıcının parolasını sahte bir parolayla değiştirmek için bu akıllı cihazda depolanan bilgileri bozabilir.
- **Servis dışı bırakma saldırıları (Denial-of-Service attack):** Servis Dışı Bırakma (DoS) saldırısı, bir sistemin veya ağın beklenen işlevini gerçekleştirmesini engelleyen herhangi bir olaydır. Bir DoS saldırısı, yazılım hataları, donanım arızaları, çevre koşulları veya kaynakların tükenmesi gibi birçok faktörden kaynaklanabilir [56].
- **Ayrıcalıklı üye saldrısı (Privileged-insider attack):** Bu tür bir saldırıda, kuruluş içindeki güvenilir bir kullanıcı, içerideki ayrıcalıklı saldırgan olarak davranabilir. İçeriden bilgi sahibi bir kullanıcı, bir kimlik doğrulama şemasının kayıt işlemi sırasında kaydolan kullanıcının gizli bilgilerini alabilir ve ardından bu bilgileri kötüye kullanmaya çalışabilir. Bu nedenle, IoT ortamı için tasarlanan güvenlik protokolleri için; herhangi bir kullanıcının oturum açma ayrıntılarının, güvenlik konusunda kuruluşun herhangi bir ayrıcalıklı üyesi veya sistem yöneticisinden de ödün vermemesi gerektiği durumu sağlanmalıdır.
- **Taklit etme saldırısı:** Bir taklit etme saldırısında, saldırgan, gönderen bir varlık adına bir ağdaki diğer alıcı varlıkları dolandırmak için sahte bir mesajla dolandırıcılık yapmaya çalışabilir. Böyle bir saldırıda alıcı, mesajın meşru bir kuruluştan alındığına inanacaktır.

- **Algılama aygıtını ele geçirme saldırılarına karşı dayanıklılık:** IoT ortamında, IoT algılama cihazları genellikle fiziksel olarak korunmaz. Dolayısıyla, algılama cihazlarının bir saldırgan tarafından fiziksel olarak ele geçirilme olasılığı vardır. Saldırgan daha sonra, ele geçirilen algılama cihazlarında depolanan bilgileri çıkararak, ele geçirilmeyen algılama cihazlarındaki iletişime zarar vermek için kullanabilir. IoT güvenlik protokollerinin, algılama cihazlarının fiziksel olarak ele geçirilmesine karşı oldukça dayanıklı olması gerekmektedir.
- **Yeni algılama cihazları yerleştirme saldırılarına karşı dayanıklılık:** IoT ortamında tasarlanan bir erişim kontrol şeması, saldırganlar tarafından yeni algılama cihazlarının konuşlandırılmasına, Sybil saldırılarına, algılama cihazlarının kopyalanmasına ve solucan deliği saldırıları gibi çeşitli saldırılara karşı dayanıklı olmalıdır. Bir solucan deliği saldırısında [57], saldırgan, bant içi veya bant dışı bir kanal yardımı ile iki uzak konum arasındaki bilgileri tünellemeye çalışır. Bir solucan deliği tüneli, iki uzak cihazın birbirine yakın olduğu izlenimini veren bir çift saldırgan düğümü tarafından oluşturulabilir. Bu nedenle, bir solucan deliği büyük miktarda ağ trafiğini aldatabilir ve atlayabilir, böylece solucan deliği cihazlarının ağ trafiğini kolayca elde etmesini ve kontrol etmesini sağlar. Solucan deliği saldırısıyla ağı dinleme, mesaj değiştirme ve paket düşürme gibi çeşitli saldırılar mümkündür [54].

  Kötü niyetli bir algılama cihazı yanlışlıkla birden fazla kimliğe büründüğünde Sybil saldırısı gerçekleşir [58,59]. Talep edilen kimlikler, mevcut algılama cihazlarının kimlikleri olabilir veya olmayabilir. Bu tür zararlı algılayıcı aygıtlar doğrudan saldırgan tarafından eklenebilir veya bu aygıtlar ayrıca saldırgan tarafından fiziksel olarak ele geçirilebilir. Bir algılama cihazı kopyalayarak çoğaltma saldırısında [60], saldırgan kasıtlı olarak, daha sonra ağa yerleştirilen, tehlikeli bir algılama cihazının birkaç kopyasını oluşturur. Saldırgan, yakalanan algılama cihazından elde edilen bilgileri kullanabilir ve bu bilgileri diğer sahte algılama cihazlarına yükleyebilir ve daha sonra bu cihazları mevcut cihazlara bağlamak için ağdaki stratejik konumlarda bu tür kötü niyetli algılama cihazlarını dağıtabilir [54].

2.1.5. IoT İçin Mevcut Platformlar

Sensörler veya RFID yongaları ile koordineli bir etkileşim yazılımı bulunması, IoT cihazlarının düzgün şekilde kurulmasını ve çalışmasını sağlamış ve onları internete

bağlamıştır. IoT cihazları için kullanılan işletim sistemleri, çeşitli yazılım ürünlerinin entegrasyonunu mümkün kılarak ağın işlevselliğini arttırmıştır ve kullanıcılara günlük işlerinde daha fazla yardımcı olmuştur [61]. Her ne kadar IoT tabanlı işletim sistemlerinde var olan güvenlik sorunları geleneksel sorunlara çok benzemese de IETF (Internet Engineering Task Force- İnternet Mühendisliği Görev Grubu) tarafından tanımlanan standartlar yine de bu tür sistemleri güvence altına almak için protokoller sunmaktadır. IoT tabanlı işletim sistemlerinden bazıları aşağıda belirtilmiştir:

i. **mbed:** Bu işletim sistemi, 32 bit Cortex-M mikro denetleyicileri ile kullanılmak üzere ARM tarafından geliştirilmiştir. C ve C ++ dilinde yazılmıştır ve Apache License 2.0 ile açık kaynaklı bir işletim sistemidir. mbed işletim sistemi, IoT cihazları için donanımsal yazılımlar geliştirmek üzere yazılım yapılandırması içermektedir [62].
ii. **RIOT:** Bu işletim sistemi INRIA, HAW Hamburg ve FU Berlin tarafından geliştirilmiştir. C ve C ++ kullanılarak yazılmıştır ve LGPL v2.1 ile açık kaynaklı bir işletim sistemidir. ARM Cortex-M3, M4, ARM7 ve AVRAtmega cihazlarıyla uyumludur. SDK yapılandırması C ve C++ uygulamalarını desteklemektedir [63].
iii. **Contiki:** Bu işletim sistemi başlangıçta Adam Dunkels tarafından yazılmıştır, ancak daha sonra Cisco, SAP ve Oxford Üniversitesi gibi çeşitli organizasyonlar tarafından geliştirilmiştir. Bu işletim sistemi, kaynakları kısıtlayan cihazlar için verimli bir şekilde çalışır. Taklit düğümler, Cooja düğümleri ve Java düğümleri olmak üzere üç tür Contiki düğümü vardır [64].
iv. **Nano-RK:** Kablosuz sensör ağları ile çalışmak üzere tasarlanan bu işletim sistemi C dili ile yazılmıştır ve açık kaynaklıdır. Carnegie Mellon üniversitesinde geliştirmiştir. Eclipse IDE bu işletim sistemi için uygulama geliştirmeyi desteklemektedir [65].
v. **Android:** Android ilk olarak Android Inc. tarafından geliştirilen ve daha sonra Google tarafından satın alınan açık kaynak kodlu mobil işletim sistemidir [66]. Linux tabanlı olan Android işletim sisteminde, uygulamalar Java diliyle yazılmaktadır. Ancak yerel kodlar ve kütüphaneler C/C++ dilleriyle yazılmıştır [67].

2.2.SALDIRI TESPİT SİSTEMLERİ

Bir ağda veya bir sistemde, her türlü yetkisiz veya onaylanmamış etkinliğe saldırı denir. Saldırı, bir ağda pasif veya aktif olarak elde edilen istenmeyen bir faaliyettir [68]. İnternet solucanları ve virüsleri şeklindeki saldırılar birkaç dakika içinde dünyaya

yayılabilmektedirler. Bir ağ, anında solucan ve virüs tehditlerini tanımalı ve azaltmalıdır. Güvenlik duvarları çok fazla şey yapabilir fakat kötü amaçlı yazılımlara ve sıfırıncı gün saldırılarına karşı koruma sağlayamazlar [69].

Sıfırıncı gün saldırıları, yazılım firmaları tarafından bilinmeyen veya açıklanmayan güvenlik açıklarından yararlanmaya çalışan bilgisayar saldırılarıdır. Sıfırıncı saat terimi ise, istismarın keşfedildiği anı tanımlar. Yazılım firmalarının bir yamayı geliştirmesi ve yayması gereken süre boyunca, ağ bu istismarlara karşı savunmasızdır. Bu hızlı hareket eden saldırılara karşı savunma yapmak, ağ güvenliği uzmanlarının; ağ mimarisinin daha karmaşık bir görüntüsünü benimsemelerini gerektirir. Günümüzde saldırıları ağdaki birkaç noktada toplamak mümkün değildir [69].

Solucan ve virüslerin bir ağa girmesini önlemeye yönelik bir yaklaşım, yöneticinin ağı sürekli izlemesi ve ağ aygıtları tarafından oluşturulan günlük dosyalarını analiz etmesidir. Bu çözüm yeterince ölçeklenebilir değildir. Günlük dosyası bilgilerini manuel olarak analiz etmek zaman alıcı bir iştir ve bir ağa karşı başlatılan saldırıların sınırlı bir görünümünü sağlayabilir. Kayıtlar analiz edildiğinde, saldırı çoktan başarılı olmuş olabilir [69].

Yıllar geçtikçe bilgisayar saldırıları daha yaygın hale gelmiştir. Sadece internete bağlı bir bilgisayar veya yerel ağa sahip olmak; saldırganların izinsiz giriş denemelerine, kötü amaçlı araçların ve programların kurulmasına ve muhtemelen onları uzaktan kontrol etmek amacıyla internetteki makineleri hedef alan sistemler kurmalarına yeterli olmaktadır [70].

Son zamanlardaki istismarlar, elde edilen bilgilerin ne kadar hassas olursa, hedef olma olasılığının da o kadar yüksek olduğunu gösteriyor. Saldırganlar şirketlere, bankalara, kamu kurumlarına vb. maddi kayıplar verebiliyor. Bazı durumlarda saldırganlar hassas bilgileri çalıyor ve üçüncü şahıslara satış yapmakla tehdit ederek şirketlere şantaj yapmaya çalışıyorlar [70], [71].

Amerika'nın en büyük bankası olan JP Morgan, 2014 yılında 76 milyon üyeyi etkileyen siber saldırılara maruz kalmıştır. 2014'te, çocuklar için bir Japon Eğitim Şirketi olan Benesse, üçüncü taraf bir ortağın eski çalışanının, reklam verenlere 28 milyona kadar müşteri hesabını ifşa ettiği büyük bir ihlal yaşadı [72]. "Sony Pictures hack" bir şirketin güvenlik ihlali sonrasında zararlarının ne kadar önemli olabileceğini ortaya çıkardı. Bu olay sebebiyle ağ sunucuları geçici olarak kapatıldı [72]. Siber güvenlik uzmanları Sony'nin 100 milyon dolara kadar zarara uğradığını tahmin etmektedir [71], [73]. Sony kanatları altındaki diğer şirketler

de çeşitli saldırılara maruz kaldılar [74]. Bilgisayar saldırılarındaki bu artış eğilimini ele almak ve tehdide cevap vermek için, endüstri uzmanları ve akademisyenler ağı izleyen ve ağdaki izinsiz faaliyetler için uyaran sistemler geliştirmek için bir araya geldiler. Bu sistemler en iyi şekilde saldırı tespit sistemleri olarak tanımlanmaktadır [70]. Saldırı Tespit Sistemi (STS) saldırıları tanımlamaya, değerlendirmeye ve raporlamaya yardımcı olan araçların, yöntemlerin ve kaynakların toplamıdır [68].

Bir saldırı tespit sistemi, bir dizi eylemin saldırı olarak kabul edilip edilmeyeceğini belirleyerek çalışır. Bir model, iyi veya kötü durumları veya eylemleri (potansiyel saldırıları) listeler [75].

Heady ve diğ. [76], saldırıyı; bir kaynağın bütünlüğüne, gizliliğine veya erişilebilirliğine müdahale girişimleri yapan bir dizi eylem olarak açıklar. Genel olarak saldırı tespit uygulaması, bir bilgisayar sisteminde gerçekleşen önemli olayların izlenmesini ve ardından saldırıların potansiyel varlıkları için analiz edilmesini içerir [77]. Alessandri, saldırıları ve izinsiz girişleri belirterek yanlış kullanım tespiti ve anomalilerin kullanımı ile güvenlik ihlaline yol açabilecek hata tespitinde kullanılan pratiklerin ve mekanizmaların toplamı olarak tanımlayarak daha kapsamlı bir saldırı tespit tanımı verir [76].

Bir STS'nin amacı, şüpheli faaliyetlerin yöneticilerini uyarmak ve hatta bazı durumlarda saldırıları engellemeye çalışmaktır [70]. STS'lerde kullanılan uygulamalar, bilgisayar sistemini korumayı amaçlayan güvenlik duvarları, erişim kontrolü veya şifreleme gibi diğer güvenlik tekniklerinden farklıdır. Bununla birlikte, bu güvenlik uygulamalarının bir sistemin savunmasını güçlendirdiği ve bir sistemin daha geniş bir kapsamının korunmasını sağladığı için birbirleriyle birlikte kullanılması şiddetle tavsiye edilir [77].

Bir güvenlik sisteminde, ilk savunma hattı "Saldırı Önleme", saldırılar engellenemezse; ikinci savunma hattı, ağ üyeleri tarafından gerçekleştirilen bir ağdaki şüpheli davranışların tespiti yani "Saldırı Tespiti" devreye girer [68].

Herhangi bir güvenlik planında, Saldırı Tespit Sistemleri (STS'ler), diğer destek sistemlerine; saldırganın tanımlanması, saldırganın bulunduğu yer, saldırı saati, saldırı faaliyeti, saldırı tipi, saldırının gerçekleştiği katman gibi bilgilerin bir kısmını veya tamamını sağlar. Saldırganla ilgili elde edilen bu özel bilgiler, saldırıların azaltılmasında (yani üçüncü savunma hattında)

ve saldırı sonuçlarının giderilmesinde çok yararlı olacaktır. Bu nedenle, saldırı tespit sistemleri ağ güvenliği için oldukça büyük bir öneme sahiptir [68].

Saldırı tespiti, tipik olarak, bir sistem veya cihazın etrafına kurulan genel koruma sisteminin bir parçasıdır ve bağımsız bir koruma önlemi değildir [78]. [79] 'da saldırı "bir kaynağın bütünlüğünü, gizliliğini veya kullanılabilirliğini tehlikeye atmaya çalışan tüm eylemler dizisi" olarak tanımlanır ve şifreleme, kimlik doğrulama, erişim kontrolü, güvenli yönlendirme vb. saldırı önleme teknikleri saldırılara karşı ilk savunma hattı olarak sunulmuştur [68].

Bununla birlikte, her türlü güvenlik sisteminde olduğu gibi, saldırılar tamamen önlenebilir değildir. Bir düğüme yapılan saldırının başarılı olması, güvenlik anahtarları gibi gizli bilgilerin saldırganların eline geçmesine yol açar. Bu da önleyici güvenlik mekanizmasının başarısızlığına neden olur. Bu nedenle, STS'ler, güvenli sistem kaynaklarını açıklamadan önce saldırıları ortaya çıkarmak için tasarlanmıştır. Her zaman güvenlik açısından ikinci bir savunma duvarı olarak kabul edilen STS'ler, bugün fiziksel güvenlik sistemlerinde kullanılan hırsız alarmlarının siber eşdeğeridir [80]. [79] 'da belirtildiği gibi, STS'lerin beklenen operasyonel gereksinimi: “anomaliler olarak tespit edilen normalite değişikliklerinin yüzdesi olarak hesaplanan düşük yanlış pozitif oran ve tespit edilen anomalilerin yüzdesi olarak hesaplanan yüksek gerçek pozitif oran” olarak verilmiştir [68].

Bir ağdaki trafiği pasif olarak izlemek için Saldırı Tespit Sistemleri (STS'ler) kullanılmaktadır. STS’ler çevrimdışı çalışarak, virüsleri kontrol eden yazılımlara benzer şekilde; yakalanan trafik akışını bilinen kötü amaçlı imzalarla karşılaştırırlar. Çevrimdışı çalışmak şu anlamlara gelir:

- STS pasif çalışır.
- STS cihazı ağda fiziksel olarak konumlandırılmıştır, böylece trafiğe ulaşmak için bu trafik yansıtılmalıdır.
- Ağ trafiği, yansıtılmadıkça STS'den geçmez.

Trafik izlenip rapor edilmesine rağmen, STS tarafından paketler üzerinde herhangi bir işlem yapılmamaktadır. Bu çevrimdışı STS uygulamasına karışık mod (promiscuous mode) denir.

Trafiğin bir kopyasıyla çalışmanın avantajı, STS'nin iletilen trafiğin paket akışını olumsuz etkilememesidir. Trafiğin bir kopyasında çalışmanın dezavantajı, STS'nin kötü amaçlı tek paketli saldırıların saldırıya yanıt vermeden önce hedefe ulaşmasını engelleyememesidir. Bir

STS genellikle bir saldırıya yanıt vermek için yönlendiriciler ve güvenlik duvarları gibi diğer ağ aygıtlarının yardımına ihtiyaç duymaktadır.

STS'lerden daha iyi bir çözüm, bir saldırıyı anında tespit edip durdurabilecek bir cihaz kullanmaktır. Saldırı Engelleme Sistemleri (Intrusion Prevention System- SES) bu işlevi yerine getirir. STS ve SES arasındaki en büyük fark, bir SES'in derhal yanıt vermesi ve herhangi bir kötü amaçlı trafiğin geçmesine izin vermemesi, bir STS'in ise kötü niyetli trafiği ele almadan önce, geçmesine izin vermesidir [69].

Hızlı hareket eden ve gelişen saldırılara karşı koyabilmek için bir ağ mimarisi modeli kullanmak gerekir. Bu model, Saldırı Tespit Sistemleri (STS) veya Saldırı Engelleme Sistemleri (SES) gibi algılama ve engelleme sistemlerini içermelidir. Ağ mimarisi bu çözümleri ağın giriş ve çıkış noktalarına entegre eder.

STS veya SES'i uygularken, mevcut sistem türlerine, bilgisayar tabanlı ve ağ tabanlı yaklaşımlara, bu sistemlerin yerleştirilmesine, imza kategorilerinin rolüne ve bir saldırı tespit edildiğinde yönlendiricilerin gerçekleştirebileceği olası işlemlere aşina olmak önemlidir.

STS ve SES teknolojileri, çeşitli özellikleri paylaşmakta ve sensörler olarak kullanılmaktadırlar. Bir STS veya SES sensörü; SES yazılımı ile yapılandırılmış bir yönlendirici, STS veya SES hizmetleri sağlamak için özel olarak tasarlanmış bir cihaz, uyarlanabilir bir güvenlik cihazına, anahtar veya yönlendiriciye takılı bir ağ modülü gibi birkaç farklı cihaz biçiminde olabilir.

STS ve SES teknolojileri, ağ trafiğindeki kalıpları tespit etmek için imzalar kullanır. İmza, bir STS veya SES'nin kötü niyetli etkinliği tespit etmek için kullandığı bir kurallar kümesidir. İmzalar, ciddi güvenlik ihlallerini tespit etmek, yaygın ağ saldırılarını tespit etmek ve bilgi toplamak için kullanılabilir. STS ve SES teknolojileri, atomik imza modellerini (tekli paket) veya bileşik imza modellerini (çoklu paket) algılayabilir.

2.2.1. Saldırı Tespitinin Tarihi

Başlangıçta saldırı tespiti, sistem yönetimi tarafından elle gerçekleştirilmekteydi. Sistem yöneticileri herhangi bir anomaliyi tanımlayan bir konsoldaki her aktiviteyi detaylı bir şekilde izlemek üzere görevlendirilmişlerdi. Bu erken kimlik formu ürettiği hatalardan dolayı etkisiz

kalmıştır. Ardından geliştirilen otomatik günlük dosyası okuyucular, usulsüzlüklerin ve yetkisiz personelin hızlıca aranmasını sağlamaktaydı [76]. İlk kimlik sürümlerinin az sayıda kuruluşa ait olduğunu, bilgisayar kullanımının yaygın bir uygulama olmadığını ve teknolojik bilgisayar kullanım çağının doğmadığını belirtmekte fayda vardır [76]. Denetim kayıtlarının tanıtılması, kimliğin adli bir tekniğe gösterilmesine yardımcı olmuştur; bu nedenle yönetim bilgi toplamış ve saldırı sürecinde değil de sadece olaylar gerçekleştikten sonraki sorunları tespit etmiştir. 90'lı yıllardan önce saldırıların analizi ve sistem yapısındaki değişiklikler, saldırı gerçekleştikten uzun süre sonra tanımlanırdı. Süreçler sıkıcıydı, çok zaman alıyordu ve ağır katılım nedeniyle insan hatası potansiyeli ortaya konuluyordu [81]. 80-90'lar boyunca araştırmalar, mevcut yazılımları güçlendirmek amacıyla yapıldı. Bu atılımın 90'larda, Denning'in [82] önerdiği Saldırı Tespit Sistemi'nin bir sonucu olarak geldiği de öne sürülmektedir. Araştırmacılar, denetim verilerini üretirken gözden geçiren bir STS geliştirdiler, bu gelişme, gerçek zamanlı STS'lerin gerçek zamanlı müdahale yöntemleri ile saldırı önlenmesine izin veren ilk versiyonunu ortaya koydu [83]. Dünya teknolojik çağa girerken, bilgi teknolojileri güvenliğine yönelik piyasa talebi arttı; STS'ler daha da geliştirildi ve büyük kuruluşlara sunuldu. Çeşitli yeni uyarı yöntemleri, saldırı deseni tanımlarındaki güncellemeler, özel kullanıcı dostu arayüzler ve tanımlandığında otomatik olarak atakları durduran önleme teknikleri gibi yeni özellikler geliştirildi [81].

Odaklar günümüzde güvenlik önlemlerinin arttırılmasına doğru yönelirken, yeni saldırı teknikleri ağın her köşesinden ortaya çıkmaya devam etti. Bu nedenle, sürekli değişen bir ortamda, tehditler sistemlere nüfuz etmenin yeni yollarını bulmak için yöntemlerinde daha fazla çeşitlilik gösterdiğinden dolayı her zaman iyileştirilmeye ve ilerlemeye devam etmeleri gerektiği geliştiriciler tarafından açıkça görülmüştür [70].

2.2.2. STS Gereksinimleri

Tasarlanması planlanan STS'ler aşağıdaki gereksinimleri karşılamalıdır [68]:

- Sisteme yeni zaaflar getirmemek,
- Çok az sistem kaynağına ihtiyaç duymak ve genel sistem performansını düşürmemek,
- Sürekli olarak çalışabilmek,
- Sisteme ve kullanıcılara karşı şeffaf olmak,
- İşbirlikçi ve şeffaf olmak için standartları kullanmak,

- Algılama aşamasında güvenilir olmak, yanlış pozitifleri ve yanlış negatifleri en aza indirmek.

2.2.3. STS'de Karar Verme

STS'ler için iki tür karar verme mekanizması vardır [68]:

- **İşbirlikçi karar verme:** Bir ağın üyelerinin tümü (veya bir kısmı) bir olayla ilgili karar vermede iş birliği yapar. Örneğin, oy çokluğu durumunda, nihai karar, iki karardan herhangi biri ile sona eren üyelerin çoğunluğu lehine verilir: "olay bir saldırıdır" veya "olay bir saldırı değildir".
- **Bağımsız karar verme:** Her üye, kendisini çevreleyen olaylara ilişkin bir karar verir.

[80] 'ye göre, bir STS, bir olayla ilgili karar verme sürecinin bir sonucu olarak aşağıda belirtilen dört karardan herhangi birini (sıfır olmayan olasılıklarla) sonuçlandırır:

- **Yanlış-negatif:** Sisteme bir saldırı var, ancak STS bunu tespit edemiyor ve olayı anormal olmayan bir sonuç olarak kabul ediyor.
- **Yanlış-pozitif:** Sisteme saldırı yok, ancak STS yanlışlıkla normal bir olayı anormal olarak değerlendiriyor.
- **Doğru-negatif:** Sisteme saldırı yok ve STS olayı anormal olmayan bir sonuç olarak kabul ediyor.
- **Doğru-pozitif:** Sisteme bir saldırı var ve STS olayı anormal bir olay olarak sonlandırıyor.

2.2.4. Saldırı Yanıtı

Saldırı mümkün olduğunda, önleyici kısma yani Saldırı Önleme Sistemine (Intrusion Prevention System-SES) bırakıldığından, STS önleyici tedbirler almaz. STS, SES'nin proaktif yoluyla karşılaştırıldığında reaktif bir şekilde çalışır [68].

Saldırı alarmı STS tarafından ne zaman üretilirse, sistem özelliklerine göre aşağıdaki eylemler gerçekleştirilir [68]:

- Bir denetim kaydı oluşturulur.

- Tüm ağ üyeleri, sistem yöneticisi (varsa) ve baz istasyonu (varsa) saldırı konusunda uyarılır. Mümkünse, saldırganın yeri ve kimliği uyarı mesajında bulundurulur.
- Varsa, saldırıyı durdurmak için bir azaltma yöntemi başlatılır. Örneğin, ağ üyelerinin ortak bir hareketi aracılığıyla otomatikleştirilmiş bir düzeltici eylem yapılabilir.

2.2.5. Saldırı Tespit Teknikleri

Saldırı tespiti için anomali tabanlı ve imza tabanlı olmak üzere iki ana teknik uygulanmaktadır.

2.2.5.1.Anomali Tabanlı Saldırı Tespit Teknikleri

Anomali tabanlı saldırı tespiti için; anomali saptama ve davranış temelli isimleri de kullanılmaktadır [70]. Anomali tabanlı bir sistem karar verme sürecini belirlemek için bir sistemin veya kullanıcının normal profilini kullanır [83]. Gelişme, dedektörün söz konusu gözlemlenen nesne (uygulama, kullanıcı, kaynak kullanımı vb.) için normal olan davranışa dair bir yargı oluşturduğu noktada başlar, sonra bu aktivitenin bir yüzdesi şüpheli ve korunmuş bir eylem olarak işaretlenebilir ve daha sonra ele alınır [84]. Bu tür bir sistem, daha önce bilinmeyen saldırıların tespiti için uygundur, çünkü saldırı hakkında önceden bilgi edinmeden çoğu saldırıları tespit eder [75], [85]. Bununla birlikte, hata teşhisi gibi sorunlar hala devam etmektedir, ayrıca bu sistemin yüksek hatalı pozitif oranlar getirdiği belirtilmektedir [83]. Bu algılama yöntemi, yanlış kullanılan dedektörlerin imzalarını tanımlamak için kullanılabilir ve bir imza tabanlı sistem ile birleştirildiğinde hibrit bir sistem oluşturabilir [82].

2.2.5.2.İmza Tabanlı Saldırı Tespit Teknikleri

İmza tabanlı saldırı tespiti için alternatif olarak; kötüye kullanım tabanlı, bilgi temelli, görünüme göre tespit gibi isimler de kullanılmaktadır. İyi bilinen saldırı modellerine dayanmaktadırlar. İmza Temelli Tespit Sistemleri, bilinen izinleri bulmak ve eşleştirmek için bir sistemde bilinen saldırı kalıplarını veya bilinen kritik noktaları kullanarak işlev görmektedirler. Alınan kararlar, saldırı süreci ve gözlemlenen sistemde neyin izlenmesi gerektiği ile ilgili bir model bilgisi temelinde oluşturulur. İmza Temelli STS'ler, daha fazla hassasiyet sunar ve tanınmış saldırıların çeşitlemelerini etkili bir şekilde tespit edebilir. Ayrıca, bir alarm, alarmın sebebi hakkında ayrıntılı bilgi verilerek tetiklendiğinde daha

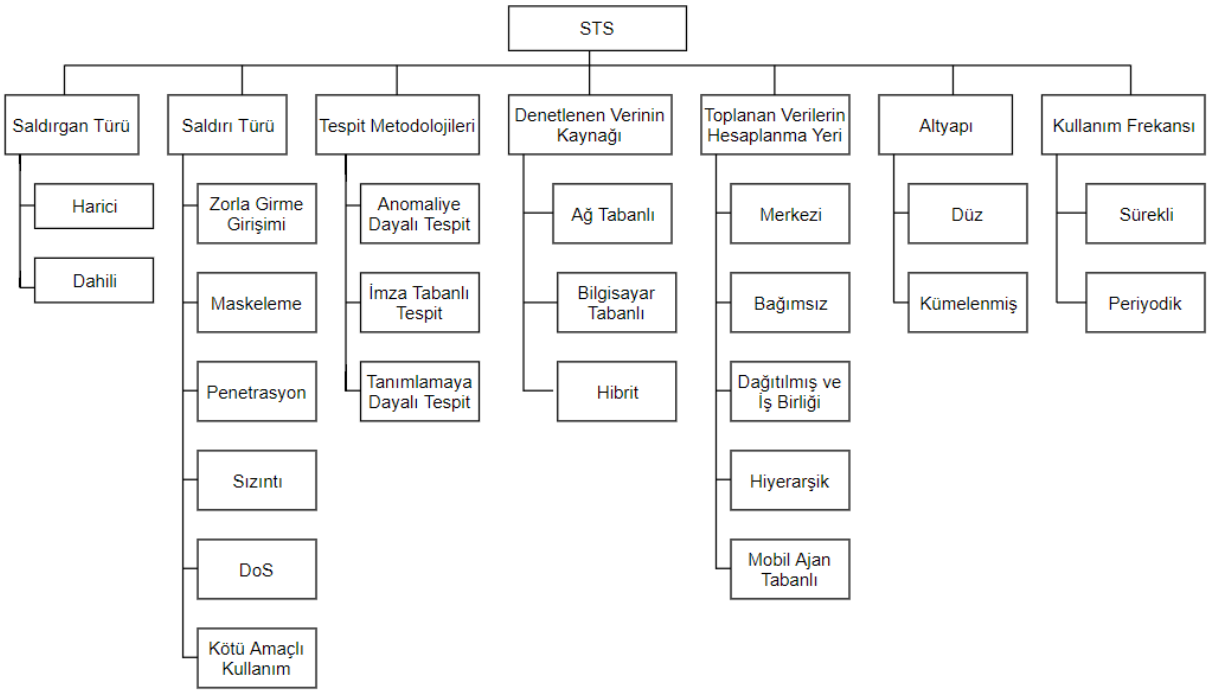
anlamlı saldırı teşhisi de sunarlar [76]. Bununla birlikte, bu algılama sistemi, kullanılamayan yeni saldırıları ve imzaları saptama kabiliyetinden yoksun olma gibi zayıflıklara sahiptir [83]. Bunun aksine, birçok ticari sistemin imza temelli tespit sistemleri kullandığı belirtilmelidir [70].

2.2.6. STS'lerin Sınıflandırılması

STS'ler iki tamamlayıcı yaklaşıma dayanabilir:

i. **Önlemeye dayalı yaklaşımlar:** Kriptografik algoritmalar, dijital imzalar, özetleme fonksiyonları kullanılarak; gizlilik, kimlik doğrulama veya bütünlük gibi önemli özellikler güvence altına alınır. Bu senaryoda, STS'lerin çevrimiçi ve gerçek zamanlı çalışıyor olmaları gereklidir.

ii. **Tespite dayalı yaklaşımlar:** STS'ler, kötü niyetli faaliyetleri etkin bir şekilde tanımlayarak birinci savunma hattı olarak hizmet ederler.



Şekil 2.2: STS'lerin sınıflandırılması.

Şekil 2.2'de gösterildiği gibi, STS çözümleri aşağıdaki şekilde bölümlere ayrılmaktadır [86], [87], [88]:

2.2.6.1.Saldırgan Türüne Göre Sınıflandırma

Ağ saldırganları harici saldırgan ve dahili saldırgan olmak üzere iki şekilde sınıflandırılabilir:

i. **Harici saldırgan:** Ağa ulaşmak için çeşitli saldırı araçlarını kullanan bir yabancıyı ifade eder.
ii. **Dahili Saldırgan:** Eskiden ağın bir üyesi olan ele geçirilmiş bir düğümü ifade eder.

Bir STS hem harici hem de dahili saldırganları tespit edebilir, ancak dahili saldırganların tespit edilmesinin daha zor olduğu belirtilmelidir. Bu, dahili saldırganların, kimlik doğrulama mekanizmaları tarafından alınan önlemleri etkisiz hale getirmek için gerekli anahtarlama malzemelerine sahip olmalarından kaynaklanmaktadır.

2.2.6.2.Saldırı Türüne Göre Sınıflandırma

Bir ağa saldırılar çeşitli şekillerde gerçekleşebilir. Bunlardan bazıları aşağıdaki gibidir:

- **Zorla girme girişimi:** Ağa yetkisiz erişim sağlama denemesidir.
- **Masquerade (Maskeleme):** Saldırganın, ağa yetkisiz erişim sağlamak için sahte bir kimlik kullanması şeklinde gerçekleştirilir.
- **Penetrasyon:** Ağa yetkisiz erişim sağlanmasıdır.
- **Leakage(Sızıntı):** Ağdan istenmeyen bir bilgi akışını ifade eder.
- **DoS:** Ağ kaynaklarının (yani iletişim bant genişliğinin) diğer, normal kullanıcılara engellenmesidir.
- **Kötü amaçlı kullanım:** Ağ kaynaklarına kasten zarar verilmesini ifade eder.

Elbette tüm sistem yöneticileri yukarıda listelenen tüm izinleri algılayabilecek mükemmel bir STS'ye sahip olmak ister. Fakat STS'ler bu saldırılara kısmi algılama çözümü sağlayabilecek durumdadır.

2.2.6.3.Tespit Metodolojilerine Göre Sınıflandırma

STS'ler fonksiyonel olarak anomaliye dayalı tespit, yanlış kullanıma dayalı tespit ve tanımlamaya dayalı tespit olmak üzere üç gruba ayrılır.

i. **Anomaliye dayalı tespit:** Bu tespit sistemi istatistiksel davranış modellemeye dayanmaktadır. Üyelerin normal operasyonları profillenir ve normal davranıştan belirli bir sapma anomalisi olarak işaretlenir. Bu tespit türünün dezavantajı, ağ davranışı hızlı bir şekilde değişebileceğinden normal profillerin periyodik olarak güncellenmesi gerektiğidir. Bu, kaynak kısıtlı sensör düğümleri üzerindeki yükü artırabilir. [89] 'ye göre, bu model, gözlenen ağın statik davranış modellerini izlemesi şartıyla saldırıları çok doğru ve tutarlı bir şekilde (düşük yanlış pozitif ve yanlış negatif oranlar) algılar. Bu algılama türünün avantajı, bilinmeyen veya daha önce karşılaşılmayan saldırıları tespit etmek için uygun olmasıdır.

ii. **İmza tabanlı (Yanlış kullanım veya Kural tabanlı) tespit***:* Daha önce bilinen saldırıların imzaları (profilleri) üretilir ve gelecekteki saldırıların tespitinde referans olarak kullanılır. Bu tür bir algılamanın avantajı, bilinen saldırıları doğru ve etkili bir şekilde saptayabilmesidir; Bu nedenle yanlış pozitif oranı düşüktür. Dezavantajı, eğer saldırı yeni bir tür ise, kötüye kullanım tespitinin onu yakalayamayacağıdır. Sobh [86], bu sistemlerin, bilinen saldırı modellerinin çoğunu veya tümünü tespit edebilen virüsten koruma sistemlerine çok benzer olduğunu, ancak henüz bilinmeyen saldırı yöntemlerinde kullanılmadığını belirtmiştir.

iii. **Tanımlamaya dayalı tespit:** Bu sistemlerde bir program veya protokolün doğru çalışmasını tanımlayan bir dizi şartname ve kısıtlamalar tanımlanmıştır. Ardından programın tanımlanmış spesifikasyonlara ve kısıtlamalara göre yürütülmesi izlenir [87]. Tanımlamaya dayalı saldırı tespit teknikleri, yasal sistem davranışını karakterize etmek için manuel olarak geliştirilen tanımlamaları ve kısıtlamaları kullanarak hem hatalı kullanım hem de anomali tabanlı tespit tekniklerinin avantajlarını birleştirir. Tanımlama tabanlı saldırı tespit teknikleri, her ikisi de saldırıları normal bir profilden sapmalar olarak algıladığı için anomali tabanlı tespit tekniklerine benzer. Tanımlamaya dayalı algılama teknikleri, manuel olarak geliştirilen tanımlamalara ve kısıtlamalara dayandığından, yanlış alarm dereceli anomali tabanlı tespit tekniklerine kıyasla düşük yanlış alarm oranına sahiptir. Öte yandan, söz konusu düşük yanlış alarm oranını elde etmenin maliyeti, ayrıntılı özelliklerin ve kısıtlamaların geliştirilmesinin çok zaman alacağı yönündedir [90].

2.2.6.4.Denetlenen Verinin Kaynağına Göre Sınıflandırma

STS'ler, analiz edilecek verilerin kaynağına bağlı olarak üç gruba ayrılabilir:

i. **Ağ tabanlı Saldırı Tespit Sistemleri (Network based Intrusion Detection System-NIDS):** NIDS, ağ iletimlerini pasif olarak veya aktif olarak dinler, iletilen paketleri yakalar ve inceler. NIDS paketin tamamını, paket içindeki yükü, IP adreslerini veya portları analiz edebilir.

ii. **Bilgisayar tabanlı Saldırı Tespit Sistemleri (Host based Intrusion Detection System-HIDS):** Bilgisayar tabanlı saldırı tespit sistemleri; bilgisayarları, bilinen ve bilinmeyen kötü amaçlı yazılımlara karşı korumak için tasarlanmıştır. Bir HIDS, sistem konfigürasyonu ve uygulama aktivitesi hakkında detaylı izleme ve raporlama yapabilir. Günlük analizi, olay korelasyonu, bütünlük kontrolü, politika uygulama ve uyarı sağlayabilir [69]. Bilgisayardaki kritik sistem dosyalarına ait değişiklikler, bilgisayara tekrar tekrar yapılan başarısız giriş denemeleri, olağandışı işlem belleği ayırmaları, olağandışı CPU veya giriş çıkış etkinliği benzeri saldırıları tespit etme yeteneğine sahiptir. HIDS bunu, bilgisayarın gerçek zamanlı sistem kullanımını izleyerek veya bilgisayardaki günlük dosyalarını inceleyerek gerçekleştirir [68].

iii. **Hibrit Saldırı Tespit Sistemleri:** Mobil ajanların kullanımıyla hem NIDS hem de HIDS bileşenlerinden verimli bir şekilde oluşur. Mobil ajanlar her bilgisayara seyahat eder ve merkezi bir ajan; anomalilerin varlığı açısından genel ağ trafiğini kontrol ederken, sistem günlük dosyası kontrolleri gerçekleştirir.

2.2.6.5.Toplanan Verilerin Hesaplanma Yerine Göre Sınıflandırma

STS toplanan verinin hesaplama yerine göre beş kategoriye ayrılır:

i. **Merkezi:** Merkezi bir bilgisayar ağdaki tüm etkinlikleri izler ve izlenen ağ etkinliği verilerini analiz ederek saldırıları tespit eder.

ii. **Bağımsız:** Bir STS her düğümde bağımsız olarak çalışır ve her karar, kendi düğümünde toplanan bilgilere dayanır. Ağın üyeleri etraflarında meydana gelen saldırıların farkında değildir, çünkü bağımsız STS'ler tek tek düğümlerin

birbirleriyle iş birliği yapmasına veya bilgi paylaşmasına izin vermez. Yalnızlarmış gibi çalışırlar.

iii. **Dağıtılmış ve İş Birliği:** Düz ağ altyapıları için önerilmektedir. Her bir düğüm, genel ağın saldırı tespitine ve yanıtına katılan bir STS ajanı çalıştırır. Bir düğüm, zayıf veya sonuçsuz kanıtlarla bir saldırı tespit ederse, bir kooperatif global saldırı tespit prosedürünü başlatabilir. Bir düğüm yerel olarak yeterli bir kanıt ile saldırı tespit ederse, bir saldırıya karşı ağı bağımsız olarak uyarabilir.

iv. **Hiyerarşik:** Bu, çok katmanlı ağ altyapıları için önerilmektedir. Küme başları üye düğümlerini izlemekten ve aynı zamanda küresel saldırı tespit kararlarına katılmaktan sorumludur.

v. **Mobil ajan tabanlı:** Her mobil ajan, seçilen bir düğümde STS'nin belirli bir görevini yerine getirmek üzere atanır ve saldırı tespiti, bu seçilen düğümlerin işbirlikçi hareketi ile gerçekleştirilir. Belirli bir süre sonra veya belirli bir görev yapıldıktan sonra, ajanlar ağ ömrünü ve/veya STS'nin verimliliğini arttırmak için önceden tanımlanmış diğer düğümlere geçebilirler.

2.2.6.6. Ağ Altyapılarına Göre Sınıflandırma

Anantvalee ve diğ. [87] STS'leri ağ altyapılarına göre düz ve kümelenmiş olarak iki gruba ayırmışlardır:

i. **Düz:** Tüm düğümler yetenekler bakımından eşit olarak kabul edilir ve yönlendirme işlevlerine katılabilirler. Bu altyapı, bir sınıfta ağ oluşturma veya konferans gibi sivil uygulamalar için uygundur.

ii. **Kümelenmiş:** Tüm düğümler eşit olarak kabul edilmez. İletim aralığındaki düğümler bir küme halinde gruplandırılır ve bu küme için yönlendirme bilgilerini merkezileştirmek amacıyla küme başı (CH) olarak bir düğüm seçerler. Genellikle, CH'ler yedek güç kaynaklı daha güçlü cihazlardan oluşur ve bu da daha uzun bir iletim menziline neden olur. Bu nedenle, CH'ler ağın sanal bir omurgasını oluşturur. Yönlendirme protokolüne bağlı olarak, ara ağ geçitleri, paketleri CH'ler arasında iletebilir. Bu tür bir altyapı modeli, bir komuta kontrol hiyerarşisine sahip askeri uygulamalar için çok uygundur.

2.2.6.7.Kullanım Frekansına Göre Sınıflandırma

Kullanım sıklığına göre STS'ler iki kategoriye ayrılır:

i. **Sürekli:** STS, ağı sürekli olarak izler.
ii. **Periyodik:** STS, ağı belirli zaman aralıklarında izler.

2.2.7. STS'lerin Avantajları ve Dezavantajları

Bir STS platformunun birincil avantajı, çevrimdışı kipte konuşlandırılmış olmasıdır. STS sensörü çevrimiçi olmadığından ağ performansını etkilemez. Gecikme gibi trafik akışı sorunlarını ortaya çıkarmaz. Ek olarak, bir sensör arızalanırsa ağ işlevselliğini etkilemez. Sadece STS'nin verileri analiz etme yeteneğini etkiler.

Bununla birlikte, bir STS platformunu kullanmanın birçok dezavantajı da vardır. Bir STS sensörü öncelikle olası olayları belirlemeye, olaylarla ilgili bilgileri kaydetmeye ve olayları raporlamaya odaklanır. STS sensörü tetikleyici paketi durduramaz ve bir bağlantıyı durduracağını garantileyemez. STS sensörleri ayrıca e-posta virüslerini ve solucanlar gibi otomatik saldırıları durdurmada daha az yardımcı olur.

STS sensörü yanıtlama eylemlerini kullanan kullanıcılar, iyi tasarlanmış bir güvenlik politikasına ve STS dağıtımları için iyi bir işlemsel anlayışa sahip olmalıdır. Kullanıcılar, beklenen saldırı tespit seviyelerini elde etmek için STS sensörlerini ayarlamaya zaman ayırmalıdır.

Son olarak, STS sensörleri çevrimiçi olmadığından, bir STS uygulaması çeşitli ağ saldırı tekniklerine karşı daha savunmasızdır [69].

3. MALZEME VE YÖNTEM

Bu tez kapsamındaki işlemler Intel(R) Core(TM) i5-8250U işlemci ve 16 GB RAM kapasitesine sahip düz üstü bilgisayar ile gerçekleştirilmiştir.

3.1.VERİ SETİ

Tez kapsamında [144]'deki makalede yer alan Mirai veri setinden 10000 satırlık bir örneklem alınarak makine öğrenmesi algoritmaları uygulanmıştır. Veri setinin ilk beş satırı Şekil 3.1'de, son beş satırı ise Şekil 3.2'de gösterildiği gibidir. Veri setinin %70'i eğitim %30'u ise test için kullanılmıştır.

	Att2	Att3	Att4	...	Att115	Att116	Durum
0	1.665584	73.208582	116.122167	...	0.0	0.0	Saldiri0
1	3.632686	77.509866	78.621641	...	0.0	0.0	Saldiri0
2	1.031044	73.999991	0.000127	...	0.0	0.0	Saldiri0
3	5.191304	81.998309	0.037190	...	0.0	0.0	Saldiri0
4	2.000099	89.998474	0.025937	...	0.0	0.0	Saldiri0

Şekil 3.1: Veri setinin ilk beş satırı.

	Att2	Att3	Att4	...	Att115	Att116	Durum
9995	116.140733	60.048285	0.699100	...	0.0	0.0	Saldiri1
9996	2.809663	135.550883	12781.827823	...	0.0	0.0	Saldiri1
9997	131.629441	60.003194	0.045115	...	0.0	0.0	Saldiri1
9998	47.367812	60.030456	0.425565	...	0.0	0.0	Saldiri1
9999	50.322459	60.028056	0.392099	...	0.0	0.0	Saldiri1

Şekil 3.2: Veri setinin son beş satırı.

Veri setinde yer alan büyük değerlerin, küçük değerlerin etkisini yok etmemesi için tüm değerler Denklem 3.1.'de gösterilen MinMaxScaler yöntemi ile 0-1 arasında ölçeklenmiştir.

$$x = \frac{x_data - \min(\text{x_data})}{\max(\text{x_data}) - \min(\text{x_data})} \tag{3.1}$$

3.2.KULLANILAN DİL VE KÜTÜPHANELER

Python: Yorumlamalı, nesneye yönelik, yüksek seviyeli bir programlama dilidir. Basit, öğrenmesi kolay sözdizimi okunabilirliği artırır ve bu nedenle program bakım maliyetini azaltır. Program modülerliğini ve kodun yeniden kullanılmasını sağlayan modülleri ve paketleri destekler. Python yorumlayıcısı ve kapsamlı standart kütüphane, tüm büyük platformlar için ücretsiz olarak mevcuttur ve serbestçe dağıtılabilir [91].

Pandas: Python programlama dili için yüksek performanslı, kullanımı kolay veri yapıları ve veri analiz araçları sağlayan açık kaynaklı bir kütüphanedir [92].

Matplotlib: Python programlama dili için grafikler üreten iki boyutlu çizim kütüphanesidir [93].

Numpy: Python ile bilimsel hesaplama için çok çeşitli veritabanlarıyla sorunsuz ve hızlı bir şekilde entegre olabilen; güçlü bir N boyutlu dizi nesnesi, gelişmiş fonksiyonlar ile C/C++ ve Fortran kodlarını entegre etmek için araçlar içeren temel bir kütüphanedir [94].

Tkinter: Tkinter modülü Tk GUI araç setinin standart Python arayüzüdür. Hem Tk hem de Tkinter, çoğu Unix platformunda ve Windows sistemlerinde mevcuttur [95].

Seaborn: Matplotlib tabanlı bir Python veri görselleştirme kütüphanesidir. Etkileyici ve bilgilendirici istatistiksel grafikler çizmek için üst düzey bir arayüz sağlar [96].

Scikit-learn: Denetimli ve denetimsiz öğrenmeyi destekleyen açık kaynaklı bir makine öğrenim kütüphanesidir. Model ayarlama, veri önişleme, model seçimi ve değerlendirmesi ve diğer birçok yardımcı program için çeşitli araçlar sağlar [97].

3.3.MAKİNE ÖĞRENMESİ

Öğrenme algoritmaları, problem çözmedeki benzersiz doğaları nedeniyle birçok uygulamada yaygın olarak kullanılmaktadır. Bu tür algoritmalar deneyimleme yoluyla otomatik olarak ilerleyen makinelerin yapımını ele alır. Son zamanlarda, öğrenme algoritmaları pratikte yaygın olarak uygulanmaktadır. Öğrenme algoritmaları; düşük hesaplama maliyetli algoritmaların ortaya çıkması, yeni algoritmaların geliştirilmesi ve büyük verilerin

kullanılabilirliği ile ilgili çalışmalara bağlı olarak gelişme göstermektedir. Makine Öğrenmesi kapsamlı ve önemli uygulamalarla son birkaç yılda büyük ölçüde ilerleme kaydetmiştir [98].

Genel olarak, öğrenme algoritmaları; bir görevi başarmak için eğitimler ve deneyimler yoluyla öğrenerek performansı arttırmayı amaçlar. Örneğin, saldırı tespitini öğrenmede görev, sistem davranışını normal veya anormal olarak sınıflandırmaktır. Sınıflandırma doğruluğu artırılarak performansta gelişme sağlanabilir. [99]

ML algoritmaları genellikle denetimli, denetimsiz ve pekiştirmeli öğrenme şeklinde kategorize edilir.

3.3.1. Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme yöntemleri, bir sınıflandırma veya tahmin modelini, öğrenilen bir planlamaya dayanarak biçimlendirir [98] ve girdi parametrelerini gözlemleyerek üretir. Başka bir deyişle, bu yöntemler giriş parametreleri ile istenen çıkış arasındaki ilişkileri yakalar. Bu nedenle, denetimli öğrenmenin ilk aşamasında, algoritmaları eğitmek için öğrenme örneklerine ihtiyaç duyulur ve bunlar yeni girdiyi tahmin etmek veya sınıflandırmak için kullanılır [100].

Denetimli öğrenmede, bilinen girdiler ve bunlara karşılık gelen çıktılar öğrenmek üzere ele alınır. Bu bilgiler, makinenin diğer girdilere ait çıktıları tanımlamasına yardımcı olur. Temel denetimli öğrenme algoritmaları şunlardır:

3.3.1.1.K-En Yakın Komşular (K-Nearest Neighbour-KNN)

Bu algoritmada, bilinmeyen bir düğümün okunması, Öklid mesafesi tarafından belirlenen en yakın komşuların ortalamasıdır [101]. Örneğin, bir WSN düğümünün okuması eksikse, komşu düğümlerin belirli bir bölgedeki ortalama okumalarından tahmin edilir. Basit hesaplamalara sahip bir algoritmadır. Bununla birlikte, büyük eğitim setleri ve yüksek boyutlar söz konusu olduğunda yanlış sonuçlar verebilir [102].

KNN parametrik olmayan bir yöntemdir. KNN sınıflandırıcıları genellikle mesafe metriği olarak Öklid mesafesini kullanır [103]. KNN sınıflandırıcısı, yeni örneği, en yakın komşularından seçilen sayının oylarına göre sınıflandırır; yani, KNN, en yakın komşularının çoğunluğunun oyuyla bilinmeyen örneklerin sınıfına karar verir. Çapraz doğrulama işlemi

sırasında farklı k değerlerinin test edilmesi, belirli bir veri kümesi için en uygun k değerini belirlemek için önemli bir adımdır. KNN algoritması basit bir sınıflandırma algoritması olmasına ve büyük eğitim veri setleri için etkili olmasına rağmen [104], en iyi k değeri daima veri setlerine bağlı olarak değişir. Bu nedenle, k'nin optimal değerinin belirlenmesi zorlu ve zaman alıcı bir süreç olabilir. KNN sınıflandırıcıları ağa saldırı tespiti ve anomali tespiti için kullanılmıştır [105-110].

3.3.1.2.Destek Vektör Makineleri (Support Vector Machine-SVM)

SVM, iki sınıf arasında bir hiper düzlem bularak sınıflandırma için kullanılır. Böyle bir düzlemi bulmak için, SVM en yakın noktalar arasındaki mesafeyi en üst seviyeye çıkarmayı amaçlar ve minimum hatayla iki sınıf arasında ayrım yapar. Sınıflandırma için doğrusal bir hiper düzlem bulunamazsa, SVM bir çekirdek fonksiyon kullanır. Bu fonksiyon, sınıfların doğrusal olarak ayrılması için yeni özellikler ekler. Yüksek doğruluk sağlaması sebebiyle SVM, IoT ve WSN'lerde güvenlik sorunlarını çözmek için yaygın olarak kullanılır [102].

SVM'ler, her bir sınıfın hiper düzlemi ve en yakın örnek noktaları arasındaki mesafe en üst seviyeye çıkarılacak şekilde iki veya daha fazla sınıf arasında veri niteliklerinde bölünen bir hiper düzlemi oluşturarak; sınıflandırma için kullanılır [111]. SVM'lerin genelleme kabiliyetleri dikkate değerdir ve çok sayıda niteliğe, az sayıda örnekleme noktasına sahip veri kümeleri için özellikle uygundur [112, 113]. Teorik olarak, SVM'ler istatistiksel öğrenmeden elde edilmişlerdir [111]. Başlangıçta, doğrusal olarak bölünebilen sınıfları, farklı sınıfların doğrusal olarak ayrılabilir (örneğin normal veya anormal) veri noktalarını içeren iki boyutlu bir düzlemde kategorize etmek için oluşturulmuşlardır. SVM'ler, hiper düzlem ile her bir sınıfın en yakın örnek noktaları arasındaki mesafeyi maksimum şekilde artırarak mükemmel bir hiper düzlem üretmelidir. SVM'lerin avantajları ölçeklenebilirlikleri, gerçek zamanlı saldırı tespitini gerçekleştirme ve eğitim modellerini dinamik olarak güncelleme yetenekleridir. SVM'ler saldırı tespiti [114-116] gibi çeşitli güvenlik uygulamalarında geniş çapta kullanılmaktadır ve bellek kullanımı açısından verimlidirler [112, 113].

3.3.1.3.Karar Ağaçları (Decision Trees-DTs)

DT tabanlı yöntemler, temel olarak örnekleri özellik değerlerine göre sıralayarak sınıflandırır. Bir ağaçtaki her köşe (düğüm) bir özelliği temsil eder ve her kenar (dal), tepe noktasının sınıflandırılması gereken bir veride sahip olabileceği bir değeri belirtir. Örnekler, başlangıç

noktasından başlayarak ve özellik değerlerine göre sınıflandırılır. Eğitim örneklerini en iyi şekilde ayıran özellik, ağacın orijin tepe noktası olarak kabul edilir [117]. DT tabanlı yaklaşımların çoğu iki ana süreçten oluşur: oluşturma (indüksiyon) ve sınıflandırma (çıkarım) [118]. Oluşturma işleminde, tipik olarak başlangıçta boş düğümleri ve dalları olan bir ağaç olarak oluşturulur. Daha sonra, eğitim örneklerini en iyi bölen özellik, ağacın orijin tepe noktası olarak kabul edilir. Bu özellik, bilgi kazanımı gibi farklı ölçümler kullanılarak seçilir. Öncelik, bir eğitim setinde sınıflar arasındaki kesişim alanını maksimum oranda azaltan ve böylece sınıflandırıcının ayırt edici gücünü artıran özellik kök düğümlerini atamaktır. Yapraklar elde edilinceye ve sınıfları belirlenene kadar her alt DT'ye aynı prosedür uygulanır. Sınıflandırma işleminde ise, ağaç oluşturulduktan sonra, bir dizi özellik ve bilinmeyen sınıfa sahip yeni örnekler, oluşturulan ağacın kök düğümleriyle başlanarak ve ağacın iç düğümlerinde özelliklerin öğrenilen değerlerine karşılık gelen yolda ilerleyerek sınıflandırılır. Bu prosedür, bir yaprak elde edilinceye kadar sürdürülür. Son olarak, yeni örneklerin ilgili etiketleri (yani, öngörülen sınıflar) elde edilir [118].

[118] 'deki araştırmacılar DT oluşumunu basitleştirmenin ana noktalarını özetlemiştir. İlk olarak, ağaç boyutunu azaltmak için ağaca ön budama veya sonradan budama uygulanır. İkincisi, aranan sınıfların alanı ayarlanır. Üçüncüsü, arama algoritması geliştirilir. Ardından, arama özellikleri sayesinde gereksiz özellikler kaldırılarak veya göz ardı edilerek veri özellikleri azaltılır. Son olarak, ağacın yapısı bir dizi kural gibi alternatif bir veri yapısına dönüştürülür. DT-bazlı yöntemlerin ana zayıflıklarından biri; yapılarının doğası gereği büyük miktarda depolamaya ihtiyaç duymalarıdır. İkinci olarak, DT tabanlı yöntemlerin anlaşılması ancak çok az sayıda DT dahil olduğunda kolaydır. Bununla birlikte, bazı uygulamalar büyük bir ağaç yapısını ve birkaç karar düğümünü içerir. Bu uygulamalarda, hesaplama karmaşıklığı yüksektir ve örnekleri sınıflandırmak için temel model karmaşıktır [118].

Bir karar ağacı, saldırı tespiti gibi güvenlik uygulamalarında diğer ML sınıflandırıcılarıyla birlikte ana sınıflandırıcı veya ortak sınıflandırıcı olarak kullanılır [119, 120].

3.3.1.4.Rassal Ormanlar (Random Forest-RF)

RF'ler denetimli öğrenme algoritmalarıdır. Bir RF'de; kesin, sağlam bir tahmin modeli ve daha iyi sonuçlar elde etmek için birkaç DT oluşturulur ve birleştirilir [121, 122]. Bu nedenle, bir RF rastgele oluşturulmuş ve bir sınıfa oy vermek için eğitilmiş çok sayıda ağaçtan oluşur.

En çok oy alan sınıf, son sınıflandırma çıktısı olarak seçilir [121]. RF sınıflandırıcısı esas olarak DT'ler kullanılarak oluşturulmuş olsa da bu sınıflandırma algoritmaları büyük ölçüde farklıdır. RF, bir sınıfı oylamaya yönelik alt kümeler oluşturmak için DT'leri kullanır; bu nedenle, sınıflandırma çıktısı sonuçların ortalamasıdır ve RF ezberlemeye (aşırı uyuma) karşı dayanıklıdır. Ayrıca, RF özellik seçimini atlar ve sadece birkaç giriş parametresi gerektirir [123]. Bununla birlikte, RF'nin kullanılması, gerekli eğitim veri setinin büyük olduğu belirli gerçek zamanlı uygulamalarda pratik olmayabilir, çünkü RF, çeşitli DT'lerin oluşturulmasına ihtiyaç duyar. RF algoritmaları ağa saldırı tespiti ve anomali tespiti için kullanılmıştır [124, 125].

3.3.1.5.Yapay Sinir Ağları (Artificial Neural Network-ANN)

ANN, sinir sistemine ve beynin çalışma sistemine benzetilerek nöronları kullanır. Biyolojiden esinlenen bu sistem farklı katmanlardan oluşur. ANN doğrusal olmayan, karmaşık sorunları çözebilir. Ancak, sahip olduğu hesaplamalar da karmaşıktır. Bu nedenle, dağıtılmış IoT ve WSN sistemlerinde kullanmak zordur [102].

3.3.1.6.Bayes Teoremi Tabanlı Algoritmalar

Bayes teoremi, olayla ilgili daha önce edinilen bilgilere dayanarak olay olasılığını açıklar [126]. Bu denetimli algoritma, sınıflandırma için büyük bir veri setine ihtiyaç duymaz. Öğrenme için olasılık dağılımlarını kullanır. Ardından, mevcut bilgilere dayanarak yeni olasılıkları elde eder. Örneğin, DoS saldırı tespiti, ağ trafiği bilgileriyle ilişkilendirilir. Bu durumda Bayes teoremini kullanmak, önceki trafik bilgilerini kullanarak ağ trafiğinin saldırı olasılığının değerlendirilmesini sağlar. Bayes sınıflandırıcıları, IoT ve WSN sistemlerinde kullanımlarını sınırlayan çevre hakkında ön bilgiye ihtiyaç duyarlar.

Bayes teoremine dayanan yaygın bir ML algoritması, Naive Bayes (NB) sınıflandırıcısıdır. NB sınıflandırıcısı, sadeliği ile bilinen yaygın olarak kullanılan denetimli bir sınıflandırıcıdır. NB, etiketlenmemiş örnek setinin belirli bir özelliğinin, özellikler arasında bağımsızlık varsayımıyla belirli bir etikete uyma olasılığını tahmin etmek için Bayes teoremini kullanır. Örneğin, saldırı tespiti için NB, trafiği normal veya anormal olarak sınıflandırmak için kullanabilir. Bağlantı süresi, bağlantı protokolü ve bağlantı durumu gibi trafik sınıflandırması için kullanılabilecek özellikler birbirine bağlı olmasına rağmen, bağımsız olarak NB sınıflandırıcısı tarafından değerlendirilir. NB sınıflandırmasında, tüm özellikler bireysel

olarak trafiğin normal veya anormal olma ihtimaline katkıda bulunur. NB ağa saldırı tespiti [127, 128] ve anomali tespiti [129, 130] için kullanılmıştır. NB sınıflandırıcılarının ana avantajları basitliği, uygulama kolaylığı, ikili ve çok sınıflı sınıflandırmaya uygulanabilirliği ve düşük eğitim örneği gereksinimidir [131]. Bununla birlikte, NB sınıflandırıcıları, özellikler arasındaki ilişkilerden ve etkileşimlerden yararlı ipuçları elde edemezler. Özellikler arasındaki etkileşimler, özellikle de özellikler arasındaki etkileşimlerin sınıflandırıcının sınıflar arasındaki kategorize etme gücünü artırmasına önemli ölçüde yardımcı olabileceği karmaşık görevlerde, doğru sınıflandırma için önemli olabilir [132].

3.3.2. Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz bir öğrenme algoritmasının amacı, girdi verilerini aralarındaki benzerliği inceleyerek farklı gruplara ayırmaktır [99]. Denetimsiz öğrenmede çıktı verilmez. Öğrenmede sadece girdiler kullanılır. Bu girdilere dayanarak, sistem bunları küme adı verilen gruplar halinde sınıflandırır. Daha sonra yeni gelen girdiler doğru gruba kategorize edilir. Bazı denetimsiz öğrenme algoritmaları şunlardır:

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

PCA veri kümelerinden önemli bilgileri alır ve yeni koordinatları tanımlamak için bunları yeni ortogonal değerler olarak tanımlar. Bu yöntem gerekli veri miktarını azaltarak büyük veri kümelerini daha küçüklerine dönüştürür [102].

PCA; büyük bir değişken kümesini, büyük kümede temsil edilen bilgilerin çoğunu koruyan azaltılmış bir kümeye dönüştürmek için uygulanabilen bir özellik azaltma tekniğidir. Bu teknik, bir dizi muhtemelen ilişkilendirilmiş özelliği, temel bileşenler olarak adlandırılan, ilişkili olmayan özelliklerin azaltılmış bir sayısına dönüştürür [133]. Bu nedenle, PCA'nın ana çalışma prensibi, IoT sistemleri için gerçek zamanlı saldırı tespitini gerçekleştirmek üzere özellik seçiminde kullanılabilir.

3.3.2.2.K-Ortalamalar Kümeleme (K-Means Clustering)

K-Ortalamalar Kümeleme, denetimsiz bir ML yaklaşımına dayanır. Bu yöntem, verilerdeki kümeleri keşfetmeyi amaçlar ve K, algoritma tarafından oluşturulacak küme sayısını ifade eder. Yöntem, her bir veri noktasının verilen özelliklere göre kümelerden birine yinelemeli

olarak dağıtılmasıyla uygulanır. Her küme benzer özelliklere sahip örnekler içerecektir. K-Ortalamalar algoritması, nihai bir sonuç elde etmek için yinelemeli geliştirme uygular. İlk önce, sentroidler tahmin edilir ve daha sonra her bir örnek Öklid mesafesine göre en yakın küme sentroidine atanır. İkinci olarak, tüm veri örnekleri belirli bir kümeye atandıktan sonra, küme sentroidleri o kümeye atanan tüm örneklerin ortalaması bulunarak yeniden hesaplanır. Algoritma, kümeleri değiştirebilecek hiçbir örnek bulunmayana kadar bu adımları yineler [134, 135]. K-Ortalamalar Kümelemesinin ana kısıtlamaları aşağıdaki gibidir:

Öncelikle, kullanıcı başlangıçta k seçmelidir. İkincisi, bu algoritma tüm küresel kümelerin yaklaşık olarak eşit sayıda örneğe sahip olduğunu varsayar. K-Ortalama algoritmaları, normal davranışı anormal davranıştan özellik benzerliği hesaplamaları ile ayırarak anomali saptamaya uygulanabilir [136, 137]. Bununla birlikte, k-ortalama kümelemenin performansı, özellikle bilinen saldırıyı tespit etmede, denetimli öğrenme yöntemlerinden daha az etkilidir [138]. Denetimsiz algoritmalar, etiketli verilerin üretilmesi zor olduğunda genellikle iyi bir seçimdir.

3.3.2.3.Birliktelik Kuralı (Association Rule -AR) algoritmaları

Eğitim veri setindeki çeşitli değişkenler arasındaki ilişkiyi araştırarak bilinmeyen bir değişkeni tanımlamak için AR algoritmaları [139] kullanılmıştır. Örneğin; X, Y ve Z; T isimli bir veri kümesinde değişkenler olsun. Bir AR algoritması, bu değişkenler arasındaki ilişkiyi, korelasyonlarını keşfetmek ve sonuç olarak bir model oluşturmak için incelemeyi amaçlar. Daha sonra, bu model yeni örneklerin sınıfını tahmin etmek için kullanılır. AR algoritmaları, saldırı örneklerinde sıklıkla birlikte bulunan değişkenlerin kombinasyonları olan sık değişken setlerini [123] tanımlamaktadır. Diğer öğrenme yöntemleriyle karşılaştırıldığında, AR yöntemleri IoT ortamlarında yaygın olarak kullanılmaz; bu nedenle, bir AR yönteminin IoT güvenliğine etkili bir çözüm sağlamak için optimize edilip edilemeyeceğini veya başka bir teknikle birleştirilip birleştirilemeyeceğini kontrol etmek için daha fazla araştırma önerilmektedir. AR algoritmalarının pratikteki ana dezavantajları şu şekildedir: İlk olarak, AR algoritmalarının zaman karmaşıklığı yüksektir. İlişkilendirme kuralları, özellikle değişkenler arasındaki frekans azaldığında yönetilemez bir miktara hızla yükselir. Verimlilik konusunu ele almak için birkaç farklı yaklaşım getirilmiş olmasına rağmen, her zaman etkili değildirler [140]. Dahası, AR algoritmaları değişkenler arasındaki basit varsayımlara dayanır. Bazı durumlarda, bu varsayımlar, özellikle saldırganların normal kullanıcıların davranışlarını taklit etmeye çalıştıkları güvenlik uygulamalarında geçerli değildir.

3.3.3. Pekiştirmeli Öğrenme (Reinforcement Learning-RL)

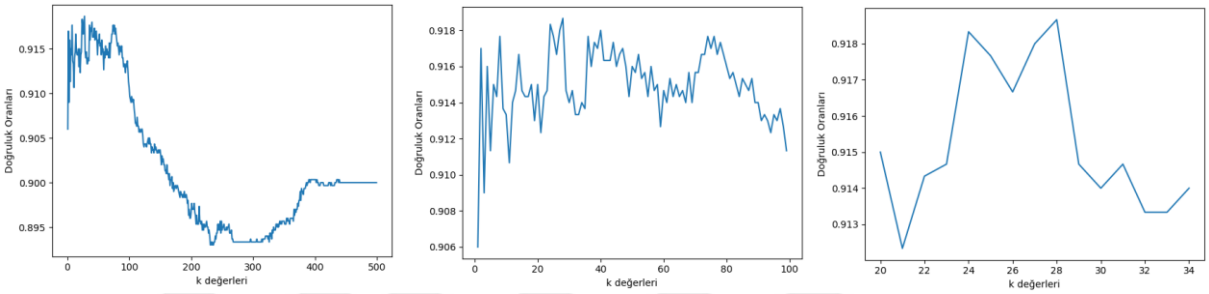
Üçüncü yaygın ML tipi pekiştirmeli öğrenmedir [141, 142]. Pekiştirmeli öğrenme algoritmaları, bir ortamdaki veriler tarafından eğitilir ve bu algoritmalar bir ortamı anlamayı, belirli bir ajan için farklı ortamlar arasından en iyi yaklaşımları keşfetmeyi amaçlar [143].

RL'deki eğitim verileri denetimli ve denetimsiz öğrenmenin yarısı kadardır. Bir eylemin doğru olup olmadığını belirtmek için RL'deki eğitim verileri; doğru eylem bulunana kadar sorun devam edecek şekilde çalışmaya devam eder [98]. Böylece, RL deneme yanılma öğrenmesi olarak kabul edilir.

Pekiştirmeli öğrenme bilinen girdilere ve bunlara karşılık gelen çıktılara sahip değildir. Bu ilişki bir ödül şeması ile öğrenilir. Görev yapmadaki performansı yüksekse, bir ödül verilir. Bu algoritma öğrenme için çevre ile etkileşime girer. Q-learning, pekiştirmeli öğrenmeye bir örnektir [102].

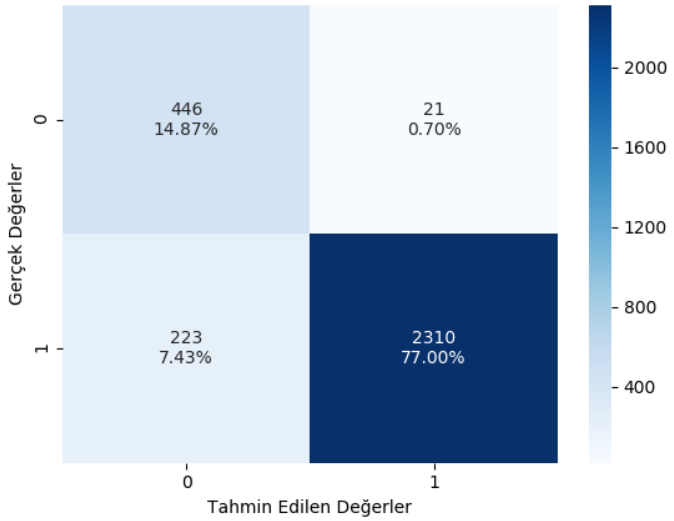
4. BULGULAR

Tez kapsamında Mirai veri setinde yer alan “Durum” sütunu baz alınarak KNN, DVM, Naive Bayes, Karar Ağaçları ve Rassal Ormanlar algoritmalarının Python dili ile uygulanması sonucu “Durum” sütunu; “saldırı” veya “saldırı değil” şeklinde sınıflandırılmıştır.



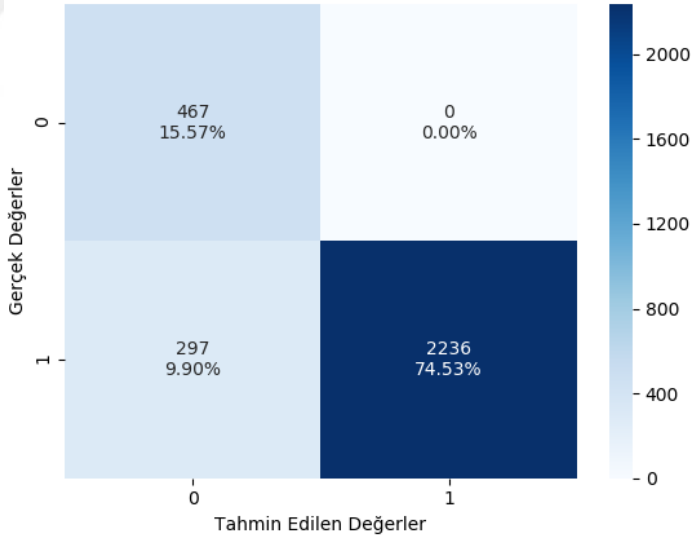
Şekil 4.1: K değerlerine karşılık gelen doğruluk oranları.

KNN algoritması için en yüksek doğruluk oranını sağlayacak K değeri Şekil 4.1’de gösterildiği gibi 28 olarak tespit edilmiştir. K değeri arttıkça performans olumsuz etkileneceğinden ve K değerinin 2 olarak belirlenmesi sonucu ortaya çıkacak doğruluk oranı ile maksimum doğruluk oranı arasındaki fark göz ardı edilebilecek kadar küçük olduğundan en uygun çözüm olarak K değerinin 2 alınması daha verimli olabilir. Fakat bu çalışmada algoritmaların maksimum doğruluk değerleri karşılaştırıldığı için K değeri uygulama içerisinde 28 olarak kullanılmıştır.



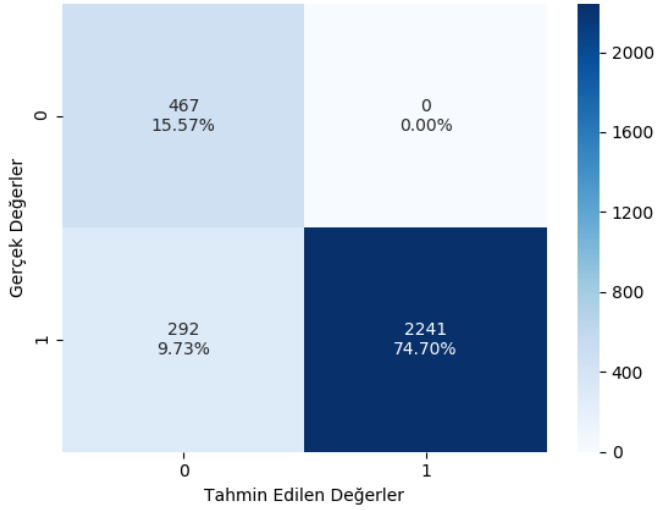
Şekil 4.2: KNN algoritması sınıflandırma sonuçları.

KNN algoritması ile test edilen 3000 veri Şekil 4.2'de gösterildiği gibi tahmin edilmiş; 2310 doğru pozitif, 21 yanlış pozitif, 223 yanlış negatif, 446 da doğru negatif olmak üzere %91,9'luk bir doğruluk oranı ortaya çıkmıştır.



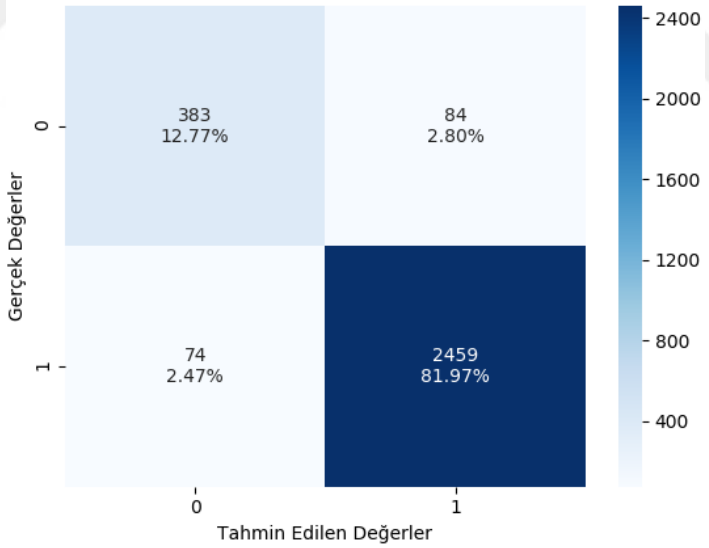
Şekil 4.3: DVM algoritması sınıflandırma sonuçları.

DVM algoritması ile test edilen 3000 veri Şekil 4.3'te gösterildiği gibi tahmin edilmiş; 2236 doğru pozitif, 0 yanlış pozitif, 297 yanlış negatif, 467 de doğru negatif olmak üzere %90,1'lik bir doğruluk oranı ortaya çıkmıştır.



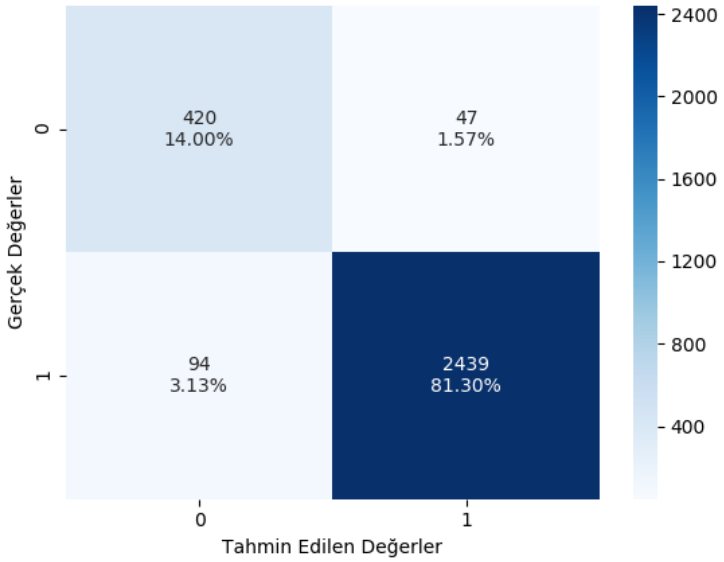
Şekil 4.4: Naive Bayes algoritması sınıflandırma sonuçları.

Naive Bayes algoritması ile test edilen 3000 veri Şekil 4.4'te gösterildiği gibi tahmin edilmiş; 2241 doğru pozitif, 0 yanlış pozitif, 292 yanlış negatif, 467 de doğru negatif olmak üzere %90,3'lük bir doğruluk oranı ortaya çıkmıştır.



Şekil 4.5: Karar ağaçları algoritması sınıflandırma sonuçları.

Karar ağaçları algoritması ile test edilen 3000 veri Şekil 4.5'te gösterildiği gibi tahmin edilmiş; 2459 doğru pozitif, 84 yanlış pozitif, 74 yanlış negatif, 383 de doğru negatif olmak üzere %94,7'lik bir doğruluk oranı ortaya çıkmıştır.



Şekil 4.6: Rassal ormanlar algoritması sınıflandırma sonuçları.

Rassal ormanlar algoritması ile test edilen 3000 veri Şekil 4.6'da gösterildiği gibi tahmin edilmiş; 2439 doğru pozitif, 47 yanlış pozitif, 94 yanlış negatif, 420 de doğru negatif olmak üzere %95,3'lük bir doğruluk oranı ortaya çıkmıştır.

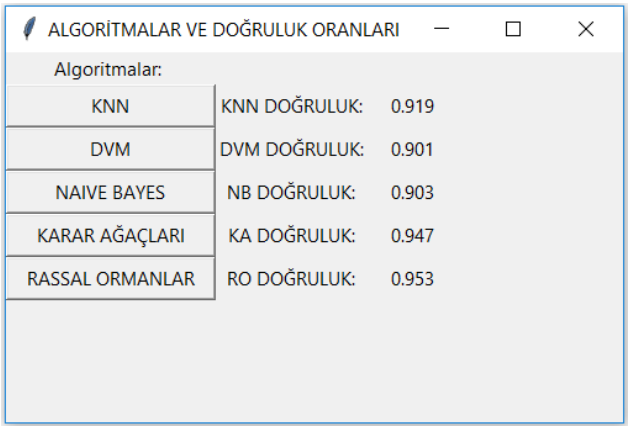
5. TARTIŞMA VE SONUÇ

Tez kapsamında incelenen algoritmaların tahmin sonuçlarının yüzdelik değerleri ve doğruluk oranları Tablo 5.1'de yer almaktadır. Rassal ormanlar ile test edilen veriler %95,3 doğruluk oranı ile en doğru tahminleri yapabilmekte iken, en düşük doğruluk oranına sahip algoritma ise %90,1 ile DVM olmuştur.

Tablo 5.1: Algoritmalar, tahmin sonuçları ve doğruluk oranları.

	Doğru Pozitif	**Yanlış Pozitif**	**Doğru Negatif**	**Yanlış Negatif**	**Doğruluk Oranı**
KNN	%77	%0,7	%14,87	%7,43	%91,9
DVM	%74,53	%0	%15,57	%9,9	%90,1
Naive Bayes	%74,7	%0	%15,57	%9,73	%90,3
Karar Ağaçları	%81,97	%2,8	%12,77	%2,47	%94,7
Rassal Ormanlar	%81,3	%1,57	%14	%3,13	%95,3

Sonuçların tez kapsamında geliştirilen uygulamaya ait bir görüntüsü de Şekil 5.1 ile gösterilmektedir.



Şekil 5.1: Uygulama ekran görüntüsü.

Elde edilen bulgularda yer alan algoritmaların sınıflandırmadaki doğruluk oranları göz önünde bulundurularak Python veya başka bir programlama dili ile anomali tabanlı STS'lerin tasarlanması, geliştirilmesi mümkündür. Bu tez ile bazı ML algoritmaları karşılaştırılmış, makine öğrenmesi kullanılarak geliştirilmek istenen STS'ler için hangi ML algoritmalarının daha iyi sonuç üreteceği gözlenmiştir. Rassal ormanlar algoritmaları doğruluk oranı en yüksek sınıflandırmayı yaparken, DVM en düşük doğruluk oranına sahip algoritma olmuştur.

ML tabanlı bir STS geliştirilmek istendiğinde algoritmaların doğruluk oranlarının dikkate alınması yanlış pozitif ve yanlış negatif uyarıları engelleyecek, geliştirilen STS'nin daha verimli çalışmasını sağlayacaktır. Fakat STS'lerde bulunması gereken diğer özellikler de göz ardı edilmemeli, algoritmaların performans karşılaştırmaları gibi metrikler de araştırılmalıdır. Doğruluk oranı yüksek olmasına rağmen saldırıların gerekli zamanda tespit edilmesini sağlayamayan algoritmalar verimli olmayacak, saldırıların hedeflerine ulaşmasını engelleyemeyecektir. Bu yüzden güvenlik sistemleri ile ilgili çözüm arayışlarında mümkün olduğunca geniş bakış açıları sunacak şekilde yeni araştırmalar yapılması elzemdir. Bu tezin sonuçları bahsi geçen bakış açılarından doğruluk oranlarını ön plana çıkarmaktadır. Fakat ileriye dönük araştırmalar kapsamında STS'ler için gerekli olan diğer kriterler de araştırılabilir ve etkili STS'lerin geliştirilmesi mümkün olabilir.

KAYNAKLAR

[1] https://static-course-assets.s3.amazonaws.com/IoE12/en/index.html

[2] A. Tewari, B.B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.027.

[3] M. Bolic, D. Simplot-Ryl, I. Stojmenovic, RFID Systems: Research Trends and Challenges, Wiley, 2010.

[4] H.-Y. Chien, SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity, IEEE Trans. Dependable Secure Comput. 4 (4) (2007) 337–340.

[5] G. Tsudik, YATRAP yet another trivial RFID authentication protocol, in: International Conference on Pervasive Computing and Communications, 2006, pp. 640–643.

[6] G. Tsudik, Family of dunces: trivial RFID identification and authentication protocols, in: Symposium on Privacy-Enhancing Technologies, 2007, pp. 45–61.

[7] M. Ohkubo, K. Suzuki, S. Kinoshita, cryptographic approach to privacyfriendly tag, in: Proc. of RFID Privacy Workshop 2003.

[8] S. Weis, S. Sarma, Ronald. Rivest, D. Engels, Security and privacy aspects of low-cost radio frequency indentification systems, in: Proc. of the 1st Security in Pervasive Computing, LNCS, 2004, pp. 201–212.

[9] A. Juels, S. Weis, Authenticating pervasive devices with human protocols, in: Victor Shoup (Ed.), Advances in Cryptology–CRYPTO'05, in: Lecture Notes in Computer Science, vol. 3126, Springer-Verlag, 2005, pp. 293–308

[10] H. Chang, A. Hari, S. Mukherjee, T.V. Lakshman, Bringing the cloud to the edge, in: Proc. IEEE Conf. INFOCOM WKSHPS, 2014, pp. 346–351

[11] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of Things, in: Proc. 1st Edition MCC Workshop Mobile Cloud Comput., 2012, pp. 13–16

[12] B. Rao, P. Saluia, N. Sharma, A. Mittal, S. Sharma, Cloud computing for Internet of Things and sensing based applications, in: Proc. 6th ICST, 2012, pp. 374–380

[13] S. Ziegler, C. Crettaz, I. Thomas, IPv6 as a global addressing scheme and integrator for the Internet of Things and the cloud, in: Proc. 28th Int. Conf. WAINA, 2014, pp. 797–802

[14] F. Li, M. Voegler, M. Claessens, S. Dustdar, Efficient and scalable IoT service delivery on cloud, in: Proc. IEEE 6th Int. Conf. CLOUD, 2013, pp. 740–747

[15] C. Wang, Z. Bi, L.D. Xu, IoT and cloud computing in automation of assembly modeling systems, IEEE Trans. Ind. Inform. 10 (2) (2014) 1426–1434

[16] B. Rochwerger, et al., The reservoir model and architecture for open federated cloud computing, IBM J. Res. Dev. 53 (4) (2009) 535–545

[17] G.C. Fox, S. Kamburugamuve, R.D. Hartman, Architecture and measured characteristics of a cloud based Internet of Things, in: Proc. Int. Conf. CTS, 2012, pp. 6–12

[18] J. Gantz, D. Reinsel, The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, in: IDC iView: IDC Anal. Future, Vol. 2007, Dec. 2012, pp. 1–16.

[19] Gartner, The Importance of 'Big Data': A Definition, June 21, 2012, https://www.gartner.com/doc/2057415/importance-big-data-definition (accessed June 12, 2014).

[20] I.S. Rubinstein, Big data: The end of privacy or a new beginning? Int. Data Privacy Law 3 (2) (2013) 74–87.

[21] M. Chen, S. Mao, Y. Liu, Big Data: A Survey, ACM/Springer Mob. Netw. Appl. 19 (2) (2014) 171–209.

[22] Y. Sun, H. Song, A.J. Jara, R. Bie, Internet of Things and big data analytics for smart and connected communities, IEEE Access 4 (2016) 766–773. http://dx.doi.org/10.1109/ACCESS.2016.2529723.

[23] Z. Lv, H. Song, P. Basanta-Val, A. Steed, M. Jo, Next-generation big data analytics: state of the art, challenges, and future research topics, IEEE Trans. Ind. Inf. (2017). http://dx.doi.org/10.1109/TII.2017.2650204.

[24] S. Fosso Wamba, S. Akter, A. Edwards, G. Chopin, D. Gnanzou, How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study, Int. J. Prod. Econ. 165 (2015) 234–246.

[25] S.K. Datta, C. Bonnet, N. Nikaein, An IoT gateway centric architecture to provide novel M2M services, in: Proc. IEEE World Forum WF-IoT, 2014, pp. 514–519.

[26] M.A. Chaqfeh, N. Mohamed, Challenges in middleware solutions for the Internet of Things, in: Proc. Int. Conf. CTS, 2012, pp. 21–26.

[27] A.P. Castellani, et al., Architecture and protocols for the Internet of Things: A case study, in: Proc. 8th IEEE Int. Conf. PERCOM Workshops, 2010, pp. 678– 683.

[28] Z. Yang, et al., Study and application on the architecture and key Technologies for IOT, in: Proc. ICMT, 2011, pp. 747–751.

[29] M. Wu, T.J. Lu, F.Y. Ling, J. Sun, H.Y. Du, Research on the architecture of Internet of Things, in: Proc. 3rd ICACTE, 2010, pp. V5-484–V5-487.

[30] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[31] https://tools.ietf.org/html/rfc6120

[32] https://tools.ietf.org/html/rfc7252

[33] http://mqtt.org/

[34] http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html

[35] https://www.omg.org/spec/DDS/1.4/PDF

[36] https://tools.ietf.org/html/rfc6762

[37] https://tools.ietf.org/html/rfc6763

[38] https://tools.ietf.org/html/rfc6550

[39] https://tools.ietf.org/html/rfc8138

[40] https://tools.ietf.org/html/rfc8137

[41] https://tools.ietf.org/html/rfc7668

[42] https://tools.ietf.org/html/rfc5134

[43] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication inLTE-advanced networks: issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[44] https://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced

[45] http://www.ieee802.org/1/files/public/docs2012/802-1-phkl-P1095-Tech-Presentation-1207-v01.pdf

[46] J. Granjal, E. Monteiro, J. Sá Silva, Security for the Internet of Things: A survey of existing protocols and open research issues, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1294–1312, 3rd Quart.

[47] IEEE, Standard, for Information, technology-Telecommunications, and information exchange between systems-Local and metropolitan area networks Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), Amendment 3: Alternative Physical Layer Extension, to support the Japanese 950 MHz bands, IEEE Std 802.15.4d-2009 (Amendment to IEEE Std 802.15.4-2006), 2009 1-27. http://dx.doi.org/10.1109/IEEESTD. 2009.4840354.

[48] Z. Yan, P. Zhang, A.V. Vasilakos, A survey on trust management for internet of things, J. Netw. Comput. Appl. 42 (0) (2014) 120–134.

[49] T. Clausen, U. Herberg, M. Philipp, A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL), in: Proc. IEEE 7th Int. Conf. WiMob, 2011, pp. 365–372.

[50] Z. Erkin, T. Veugen, R.L. Lagendijk, Generating private recommendations in a social trust network, in: Proceedings of the International Conference on Computational Aspects of Social Networks, CASoN, 2011. pp. 82–87.

[51] D. Gessner, A. Olivereau, A.S. Segura, A. Serbanati, Trustworthy infrastructure services for a secure and privacy-respecting Internet of Things, in: Proceedings of the IEEE 11th international conference on trust, security and privacy in computing and communications, TrustCom, 2012, pp. 998–1003.

[52] Y. Liu, K. Wang, Trust control in heterogeneous networks for Internet of Thing, in: Proceedings of the International Conference on Computer Application and System Modeling, ICCASM, 2010, pp. 632–636.

[53] M. Nitti, R. Girau, L. Atzori, A. Iera, G. Morabito, A subjective model for trustworthiness evaluation in the social Internet of Things, in: Proceedings of the IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC, 2012, pp. 18–23.

[54] Ashok Kumar Das , Sherali Zeadally, Debiao He, Taxonomy and analysis of security protocols for Internet of Things, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.06.027.

[55] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Proceedings of 19th Annual International Cryptology Conference, CRYPTO'99, in: LNCS, vol. 1666, Santa Barbara, California, USA, 1999, pp. 388–397.

[56] A. Wood, J.A. Stankovic, Denial of service in sensor networks, IEEE Comput. 35 (10) (2002) 54–62.

[57] Y. Hu, A. Perrig, D.B. Johnson, Pachet leashes: a defense against wormhole attacks in wireless networks, in: Proceedings of IEEE International Conference on Computer and Communications, INFOCOM'03, vol. 3, San Francisco, CA, USA, 2003, pp. 1976–1986.

[58] J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: Analysis and defenses, in: Proceedings of third IEEE International Conference on Information Processing in Sensor Networks, IPSN'04, Berkeley, California, USA, 2004, pp. 259–268.

[59] J.R. Douceur, The sybil attack, in: First International Workshop on Peer-To-Peer Systems, IPTPS'02, in: Lecture Notes in Computer Science (LNCS), vol. 2429, Springer-verlag, Cambridge, MA, USA, 2002, pp. 251–260.

[60] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: IEEE Symposium on Security and Privacy, S & P'05, Oakland, California, USA, 2005, pp. 49–63.

[61] Tuhin Borgohain, Uday Kumar, Sugata Sanyal, Survey of Operating Systems for the IoT Environment, arXiv preprint arXiv:1504.02517, 2015/4/13.

[62] Arm Limited MBed OS, Available at: https://mbed.org/technology/os/ (Last Accessed: February 2018).

[63] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, T.C. Schmidt, RIOT OS: Towards an OS for the Internet of Things, in: Proc. IEEE Conf. INFOCOM WKSHPS, 2013, pp. 79–80.

[64] A. Dunkels, B. Gronvall, T. Voigt, Contiki—A lightweight and flexible operating system for tiny networked sensors, in: Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw., 2004, pp. 455–462.

[65] Anand Eswaran, Anthony Rowe, Raj Rajkumar, Nano-rk: an energy-aware resource-centric rtos for sensor 5 networks, in: Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International, IEEE, 2005, p. 10-pp.

[66] Haya A., Ghouzali S., 2018, Android data storage security: A review.

[67] Faruki P., Bharmal A., Laxmi V., Ganmoor V., Gaur M., Conti M., Rajarajan M., 2014, Android security: a survey of issues, malware penetration and defenses, Commun. Surveys Tutorials, pp. 998-1022.

[68] I. Butun, S.D. Morgera, and R. Sankar, A Survey of Intrusion Detection Systems in Wireless Sensor Networks, IEEE Communicatıons Surveys & Tutorials, Vol. 16, No. 1, Fırst Quarter 2014.

[69] https://static-course-assets.s3.amazonaws.com/CCNAS2/en/index.html

[70] Z. Dewa, L. Maglaras, Data Mining and Intrusion Detection Systems, International Journal of Advanced Computer Science and Applications, January 2016.

[71] S Tobak. (18 December 2014). Fox Business. “3 Revelations from the Sony Hack.” Last accessed on 29 January 2015, http://www.foxbusiness.com/technology/2014/12/18/revelations-fromsony-hack/.

[72] “Report and response regarding Leakage of Customers’ personal Information.” (10 September 2014). Last accessed on 17 February 2015, http://blog.benesse.ne.jp/bh/en/ir_news/m/2014/09/10/uploads/news_20 140910_en.pdf.

[73] A, Peterson. (5 December 2014). The Washington Post. “Why it’s so hard to calculate the cost of the Sony Pictures hack.” Last accessed on 29 January 2015, http://www.washingtonpost.com/blogs/theswitch/wp/2014/12/05/why-its-so-hard-to-calculate-the-cost-of-thesony-pictures-hack/.

[74] Trend Micro Incorporated, (22 December 2014). Simply Security. “The Reality of the Sony Pictures Breach.” Last accessed on 29 January 2015, http://blog.trendmicro.com/reality-sony-pictures-breach/.

[75] R.S. Wahono, “A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks,” In Journal of Software Engineering, vol. 1, April 2015, pp. 1-16.

[76] R. Heady, G.F. Luger, A. Maccabe and M. Servilla, “The architecture of a Network Level Intrusion Detection System,” Department of Computer Science, College of Engineering, University of New Mexico, 1990, pp. 1- 17.

[77] R. Bace and P. Mell, “NIST Special Publication on Intrusion Detection Systems,” Booz-Allen and Hamilton inc, Mclean VA, 2001, pp. 5-22.

[78] M. Ngadi, A.H. Abdullah, and S. Mandala, “A survey on MANET intrusion detection”, International J.Computer Science and Security, volume 2, number 1, pages 1-11, 2008. [63] Y. Zhang, W. Lee, and Y.A. Huang, “Intrusion detection techniques for mobile wireless networks”, J. Wireless Networks, vol. 9, num. 5, pp. 545-556, 2003.

[79] Y. Zhang, W. Lee, and Y.A. Huang, “Intrusion detection techniques for mobile wireless networks”, J. Wireless Networks, vol. 9, num. 5, pp. 545-556, 2003.

[80] A. Patcha and J.M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends”, Elsevier J. Computer Networks, volume 51, number 12, pages 3448-3470, 2007.

[81] J. Allen, A. Christie, W. Fithen, J. McHugh and J. Pickel, “State of the practice of intrusion detection technologies,” vol. CMU/SEI-99-TR-028, Canergie-Mellon Univ Pittsburgh PA Software Engineering Institute, 2000 , pp. 3-23.

[82] Abhaya, K. Kumar, R. Jha and S. Afroz, “Data Mining Techniques for Intrusion Detection: A Review,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, June 2014, pp. 6938- 6941.

[83] R.A. Kemmerer and G. Vigna, “Intrusion Detection : A brief History and Overview,” Computer, 2002 [supplement to security and privacy magazine], pp. 27-30.

[84] S. Choudhury and A.Bhowal,“Comparative Analysis of Machine Learning Algorithms along with Classifiers for Network Intrusion Detection.” In: Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, IEEE, 2015, pp. 89-95.

[85] Leandros A. Maglaras, Jianmin Jiang, Tiago Cruz, “Integrated OCSVM mechanism for intrusion detection in SCADA systems”, IET Electronics Letters, Volume 50, issue 25, December 2014, p 1935-1936.

[86] T.S. Sobh, “Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art”, Elsevier J. Computer Standards and Interfaces, volume 28, number 6, pages 670-694, 2006.

[87] T. Anantvalee and J. Wu, “A survey on intrusion detection in mobile ad hoc networks”, Springer J. Wireless Network Security, pages 159-180, 2007.

[88] P. Albers, O. Camp, J. Percher, B. Jouga, L. M, and R. Puttini, “Security in Ad Hoc Networks: A General Intrusion Detection Architecture Enhancing Trust Based Approaches,” Proc. 1st International Workshop on Wireless Information Systems WIS-2002), pp. 1-12, April 2002.

[89] O. Kachirski and R. Guha, “Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks,” Proc. 36th Annual Hawaii International Conference on System Sciences (HICSS’03), p. 57.1, January 2003.

[90] B. Sun, L. Osborne, Y. Xiao and S. Guizani, “Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Networks”, IEEE Trans. Wireless Commun., vol. 14, num. 5, pp. 56-63, 2007.

[91] https://www.python.org/doc/essays/blurb/

[92] https://pandas.pydata.org

[93] https://matplotlib.org/

[94] https://numpy.org/

[95] https://docs.python.org/2/library/tkinter.html

[96] https://seaborn.pydata.org/

[97] https://scikit-learn.org/stable/getting_started.html

[98] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255-260, 2015.

[99] M. Ali Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, M. Guizani, A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security.

[100] J. Franklin, "The elements of statistical learning: data mining, inference and prediction," The Mathematical Intelligencer, vol. 27, no. 2, pp. 83-85, 2005.

[101] F. Chen, P. Deng, J. Wan, D. Zhang, A. Vasilakos, and X. Rong, “Data mining for the internet of things: literature review and challenges,” Int. J. Distrib. Sens. Netw., vol. 11, p. 431047, 2015.

[102] M. Mamdouh, M. A. I. Elrukhsi, A. Khattab, Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey, IEEE International Conference on Computer and Applications (ICCA), 2018.

[103] P. Soucy and G. W. Mineau, "A simple KNN algorithm for text categorization," in Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, 2001, pp. 647-648: IEEE.

[104] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," Neurocomputing, vol. 195, pp. 143-148, 2016.

[105] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection1," Computers & security, vol. 21, no. 5, pp. 439-448, 2002.

[106] A. O. Adetunmbi, S. O. Falaki, O. S. Adewale, and B. K. Alese, "Network intrusion detection based on rough set and k-nearest neighbour," International Journal of Computing and ICT Research, vol. 2, no. 1, pp. 60-66, 2008.

[107] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, no. 10, pp. 11994-12000, 2009.

[108] L. Li, H. Zhang, H. Peng, and Y. Yang, "Nearest neighbors based density peaks approach to intrusion detection," Chaos, Solitons & Fractals, vol. 110, pp. 33-40, 2018.

[109] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm," in Information & Communication Technology and System (ICTS), 2017 11th International Conference on, 2017, pp. 181-186: IEEE.

[110] M.-Y. Su, "Real-time anomaly detection systems for Denialof-Service attacks by weightedk-nearest-neighbor classifiers," Expert Systems with Applications, vol. 38, no. 4, pp. 3492-3498, 2011.

[111] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," Journal of machine learning research, vol. 2, no. Nov, pp. 45-66, 2001.

[112] V. Vapnik, The nature of statistical learning theory. Springer Science & Business Media, 2013.

[113] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2015.

[114] W. Hu, Y. Liao, and V. R. Vemuri, "Robust Support Vector Machines for Anomaly Detection in Computer Security," in ICMLA, 2003, pp. 168-174.

[115] Y. Liu and D. Pi, "A Novel Kernel SVM Algorithm with Game Theory for Network Intrusion Detection," KSII Transactions on Internet & Information Systems, vol. 11, no. 8, 2017.

[116] C. Wagner, J. François, and T. Engel, "Machine learning approach for ip-flow record anomaly detection," in International Conference on Research in Networking, 2011, pp. 28-39: Springer.

[117] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3-24, 2007.

[118] S. B. Kotsiantis, "Decision trees: a recent overview," Artificial Intelligence Review, vol. 39, no. 4, pp. 261-283, 2013.

[119] K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis," in SoutheastCon, 2016, 2016, pp. 1-6: IEEE.

[120] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690-1700, 2014.

[121] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[122] D. R. Cutler et al., "Random forests for classification in ecology," Ecology, vol. 88, no. 11, pp. 2783-2792, 2007.

[123] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2016.

[124] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, 2006, pp. 8 pp.-269: IEEE.

[125] Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine," in Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on, 2017, vol. 1, pp. 635-638: IEEE.

[126] G. D'Agostini, "A multidimensional unfolding method based on Bayes' theorem," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 362, no. 2-3, pp. 487-498, 1995.

[127] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," International journal of computer science and network security, vol. 7, no. 12, pp. 258-263, 2007.

[128] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, vol. 4, pp. 119-128, 2012.

[129] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," Procedia Computer Science, vol. 60, pp. 708-713, 2015.

[130] M. Swarnkar and N. Hubballi, "OCPAD: One class Naive Bayes classifier for payload based anomaly detection," Expert Systems with Applications, vol. 64, pp. 330-339, 2016.

[131] G. E. Box and G. C. Tiao, Bayesian inference in statistical analysis. John Wiley & Sons, 2011.

[132] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in Advances in neural information processing systems, 2002, pp. 841-848.

[133] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," Chemometrics and intelligent laboratory systems, vol. 2, no. 1-3, pp. 37-52, 1987.

[134] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A kmeans clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100- 108, 1979.

[135] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern recognition letters, vol. 31, no. 8, pp. 651-666, 2010.

[136] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in GI/ITG Workshop MMBnet, 2007.

[137] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," IEEE communications surveys & tutorials, vol. 16, no. 1, pp. 303-336, 2014.

[138] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?" in International Conference on Image Analysis and Processing, 2005, pp. 50-57: Springer.

[139] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in Acm sigmod record, 1993, vol. 22, no. 2, pp. 207-216: ACM.

[140] S. Kotsiantis and D. Kanellopoulos, "Association rules mining: A recent overview," GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, pp. 71-82, 2006.

[141] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, p. 529, 2015.

[142] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction (no. 1). MIT press Cambridge, 1998.

[143] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996-2018, 2014.

[144] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection http://dx.doi.org/10.14722/ndss.2018.23204

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Umur KURİŞ
Doğum Yeri	Tarsus
Doğum Tarihi	12.07.1988
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	0 546 692 1263
E-Posta Adresi	umurkuris@gmail.com
Web Adresi	-



Eğitim Bilgileri	
Lisans	
Üniversite	Karabük Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	2015

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi-Cerrahpaşa
Enstitü Adı	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı