**T.C.**
**MİLLÎ SAVUNMA ÜNİVERSİTESİ**
**HEZÂRFEN HAVACILIK VE UZAY TEKNOLOJİLERİ ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
**SİBER GÜVENLİK PROGRAMI**

# RPL TABANLI IOT CİHAZLARI ZAFİYETİNİN TESPİTİ İÇİN MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRILMASI

**YÜKSEK LİSANS TEZİ**

**MURAT UĞUR KİRAZ**
**3181110**

**TEZ DANIŞMANI: Dr.Öğr.Üyesi Atınç YILMAZ**

**İSTANBUL**
**OCAK 2022**

**T.C.**
**MİLLÎ SAVUNMA ÜNİVERSİTESİ**
**HEZÂRFEN HAVACILIK VE UZAY TEKNOLOJİLERİ ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
**SİBER GÜVENLİK PROGRAMI**

# RPL TABANLI IOT CİHAZLARI ZAFİYETİNİN TESPİTİ İÇİN MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRILMASI

**YÜKSEK LİSANS TEZİ**

**MURAT UĞUR KİRAZ**
**3181110**

**Tezin Enstitüye Verildiği Tarih: 18 Ocak 2022**
**Tezin Savunulduğu Tarih: 12 Ocak 2022**

**Tez Oy birliği / Oy çokluğu ile başarılı bulunmuştur.**

| | Unvan Ad Soyad | İmza |
|---|---|---|
| **Tez Danışmanı** | **Dr.Öğr.Üyesi Atınç YILMAZ** | |
| **Jüri Üyeleri** | **Dr.Öğr.Üyesi Ediz ŞAYKOL** | |
| | **Dr.Öğr.Üyesi Serkan AYVAZ** | |

**İSTANBUL**
**OCAK 2022**

## ÖZGÜNLÜK RAPORU

Tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve ç) Sonuç kısımlarından oluşan toplam 99 sayfalık kısmına ilişkin, 14/01/2022 tarihinde şahsım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı %9'dur.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç/dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Millî Savunma Üniversitesi Hezârfen Havacılık ve Uzay Teknolojileri Enstitüsü Lisansüstü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Usul ve Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Murat Uğur KİRAZ

18 Ocak 2022

## ETİK BEYAN

Millî Savunma Üniversitesi Enstitüleri Lisansüstü Tez Hazırlama Kılavuzu'nda yer alan kurallarına uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bu tezdeki düşünce, görüş, varsayım, sav veya tezler bana aittir; Millî Savunma Bakanlığı, Millî Savunma Üniversitesi ve Hezârfen Havacılık ve Uzay Teknolojileri Enstitüsü sorumlu tutulamaz.

Murat Uğur KİRAZ
18 Ocak 2022

*Şu ana kadar benden maddi ve manevi desteğini esirgemeyen kıymetli annem, babam ve eşime.*

## ÖNSÖZ ve TEŞEKKÜR

Bilgisayar bilimleri, Dünya üzerinde İstanbul'un Fethi, Fransız İhtilali veya sanayi devrimi gibi muazzam bir etki bıraktı, Dünya düzenini değiştirdi, Dünya'yı yeniden şekillendirdi. Bu asla yadsınamayacak bir gerçek. Bu gerçeği en iyi şekillendiren, yöneten, ilerleten kişi / kurum / devlet veya her türlü topluluk, diğerlerine nazaran her türlü avantajı elinde tutacaktır.

Şu anda günlük hayatımızda ve endüstride kullanılan ve birbiri ile iletişimde olup internete çıkan "nesnelerin interneti cihazları" ile, bilgisayar bilimleri devriminin ilerleyişi muazzam bir ivme kazandı. Özellikle enerji sorununu olduğu ve enerjiyi en etkin şekilde kullanmamız gereken günümüzde, nesnelerin interneti cihazlarının azami enerji verimini elde etmek maksadıyla düşük güçte ve kayıplı ağlarda çalışan bir protokol Mart 2012 yılında İnternet Mühendisliği Görev Gücü (IETF-Internet Engineering Task Force) tarafından RPL ile (Routing Protocol for Low-Power and Lossy Networks - Düşük Güçlü ve Kayıplı Ağlar için IPv6 Yönlendirme Protokolü) ortaya çıkarıldı.

RPL protokolü ile nesnelerin interneti cihazları kararlı, hızlı ve enerji tasarrufu sağlayan bir mahiyette çalışmaktadır, ancak bu protokolünün karmaşıklığı ve bu protokolde çalışan 6LoWPAN (IPv6 over Low -Power Wireless Personal Area Networks- Düşük Güçlü Kablosuz Kişisel Alan Ağları üzerinden IPv6) cihazlarının düşük güvenlikli doğası gereği, bu protokol ağ içindeki veya dışındaki saldırılara karşı savunmasızdır. Bu yüzden, RPL protokolünde çalışan cihazlarda yapılan saldırıların makine öğrenmesi algoritmaları ile tespiti için literatürde birçok saldırı tespit sistemi geliştirilmiştir. Geliştirilen bu sistemler arasındaki en iyi sonucu veren sistemi bulmak veya daha iyisini geliştirmek adına kendime şu soruyu sordum. RPL protokolündeki saldırıları tespit etmede en etkili makine öğrenimi algoritması hangisidir? Bu tez çalışması bu soru ile doğmuştur.

Bu tez çalışmasında, birinci bölümde, tezin amacı, literatürde olan mevcut çalışmalar ve bu tez çalışmasının literatürdeki hangi eksikleri giderdiği anlatılmıştır. İkinci bölümde, IoT cihazlarında işleyen protokoller ile ilgili genel bilgiler verilmiş, özellikle

RPL protokolü ve bu protokolde gerçekleştirilebilecek saldırılar detaylı biçimde açıklanmıştır. Üçüncü bölümde, deneylerin nasıl yapıldığı hangi yöntemlerin izlendiği açıklanmıştır. Sonuç bölümünde ise, deneyler sonucunda elde edilen veriler yorumlanmıştır.

Bu tez çalışmasında bana her türlü desteğini veren ve yol gösteren saygıdeğer öğretmenim sayın Dr.Atınç YILMAZ'a, çalışmalarım esnasında manevi desteğini esirgemeyen ve çalışmalarımda beni motive eden kıymetli eşim Başak TÜRKEN KİRAZ'a, değerli anne-babam Firdevs KİRAZ'a ve Mehmet KİRAZ'a sonsuz şükranlarımı sunarım.

İstanbul, Ocak 2022 Murat Uğur KİRAZ

# İÇİNDEKİLER

**Sayfa**

**Sayfa**

## TABLOLAR LİSTESİ

**Sayfa**

## ŞEKİLLER LİSTESİ

**Sayfa**

## KISALTMALAR

| | | |
|---|---|---|
| **6LoWPAN** | **:** | IPv6 over Low -Power Wireless Personal Area Networks (Düşük Güçlü Kablosuz Kişisel Alan Ağları üzerinden IPv6) |
| **AMQP** | **:** | Advanced Message Queuing Protocol (Gelişmiş Mesaj Kuyruk Protokolü) |
| **BLE** | **:** | Bluetooth Low Energy (Bluetooth Düşük Enerji) |
| **CoAP** | **:** | Constrained Application Protocol (Kısıtlı Uygulama Protokolü) |
| **CSMA/CA** | **:** | Carrier sense multiple access with collision avoidance (Çarpışmadan kaçınma ile taşıyıcı algılama çoklu erişim) |
| **DAG** | **:** | Directed Acyclic Graph (Yönlendirilmiş Asiklik Grafik) |
| **DAO** | **:** | DODAG Advertisement Object (DODAG Duyuru Nesnesi) |
| **DAO-ACK** | **:** | DODAG Advertisement Object Acknowledgement (DODAG Duyuru Nesnesi Kabulü) |
| **DDS** | **:** | The Data Distribution Service (Veri Dağıtım Hizmeti) |
| **DIO** | **:** | DODAG Information Object (DODAG Bilgi Nesnesi) |
| **DIS** | **:** | DODAG Information Solicitation (DODAG Bilgi İsteği) |
| **DNS-SD** | **:** | Domain Name System Service Discovery (Etki Alanı Adı Sistem Hizmet Keşfi) |
| **DODAG** | **:** | Destination Oriented DAG (Hedefe Yönlendirilmiş DAG) |
| **DRA** | **:** | Decreased Rank Attack (Azalan Rank Saldırısı) |
| **DSSS** | **:** | Direct Sequence Spread Spectrum (Doğrudan Dizi Yayılma Spektrumu) |
| **ELNIDS** | **:** | Ensemble Learning based Network Intrusion Detection System (Ensemble Learning Tabanlı İzinsiz Giriş Tespit Sistemi) |
| **EPC** | **:** | Electronic Product Code (Elektronik Ürün Kodu) |
| **FFD** | **:** | Full Functioned Devices (Tam İşlevli Cihazlar) |
| **GN** | **:** | (Gerçek Negatif) |
| **GP** | **:** | (Gerçek Pozitif) |
| **HF** | **:** | Hello Flood (Merhaba Taşma) |
| **IEEE** | **:** | Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisleri Enstitüsü) |
| **IETF** | **:** | Internet Engineering Task Force (İnternet Mühendisliği Görev Gücü) |
| **IoT** | **:** | Internet of Things (Nesnelerin İnterneti) |
| **IPv4** | **:** | Internet Protocol Version 4 (İnternet Protokolü Sürüm 4) |
| **IPv6** | **:** | Internet Protocol Version 6 (İnternet Protokolü Sürüm 6) |
| **JSON** | **:** | JavaScript Object Notation (JavaScript Nesne Notasyonu) |
| **KDE** | **:** | Kernel Density Estimation (Çekirdek Yoğunluğu Tahmini) |
| **LLN** | **:** | Low-Power and Lossy Networks (Düşük Güçlü ve Kayıplı Ağlar) |
| **LR** | **:** | Lojistik Regresyon (Lojistik Regresyon) |
| **LR-WPAN** | **:** | Low Rate Wireless Personal Area Network (Düşük Hızlı Kablosuz Kişisel Alan Ağı) |
| **LTE-A** | **:** | Long-term Evolution – Advanced (LTE-A) (Gelişmiş Uzun Vadeli Evrim) |
| **MAC** | **:** | Media Access Control (Medya Erişim Kontrolü) |
| **mDNS** | **:** | Multicast Domain Name System (Çok Noktaya Yayın Etki Alanı Adı Sistemi) |

| | | |
|---|---|---|
| **MLTKNN** | **:** | (K En yakın Komşu algoritmasına dayanan bir makine öğrenmesi algoritması) |
| **MQTT** | **:** | Message Queuing Telemetry Transport (Mesaj Kuyruğu Telemetri Aktarımı) |
| **OSI** | **:** | Open Systems Interconnection (Açık Sistem Arabağlantısı) |
| **PAN** | **:** | Personal Area Network (Kişisel Alan Ağı) |
| **RFD** | **:** | Reduced Functioned Devices (Azaltılmış İşlevli Cihazlar) |
| **RFID** | **:** | radio-frequency identification (Radyo frekansı tanımlama) |
| **RPL** | **:** | Routing Protocol for Low-Power and Lossy Networks (Düşük Güç Tüketimli ve Kayıplı Ağlar için Yönlendirme Protokolü) |
| **SVM** | **:** | Support Vector Machine (Destek Vektör Makinesi) |
| **TCP** | **:** | Transmission Control Protocol (İletim Kontrol Protokolü) |
| **UDP** | **:** | User Datagram Protokol (kullanıcı veri paket protokolü) |
| **VNIA** | **:** | Version Number Increase Attack (Sürüm Numarası Artırma Saldırısı) |
| **WPAN** | **:** | Wireless Personal Area Network (Kablosuz Kişisel Alan Ağı) |
| **XML** | **:** | eXtensible Markup Language (Genişletilebilir İşaretleme Dili) |
| **XMPP** | **:** | Extensible Messaging and Presence Protocol (Genişletilebilir Mesajlaşma ve Durum Protokolü) |
| **YN** | **:** | (Yanlış Negatif) |
| **YP** | **:** | (Yanlış Pozitif) |

# ÖZ

**RPL Tabanlı IOT Cihazları Zafiyetinin Tespiti İçin Makine Öğrenmesi Algoritmalarının Karşılaştırılması**

Murat Uğur KİRAZ

Millî Savunma Üniversitesi, Havacılık ve Uzay Bilimleri Enstitüsü

İstanbul, Ocak 2022

RPL protokolü (Routing Protocol for Low-Power and Lossy Networks-Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü), Nesnelerin İnterneti (IoT-Internet of Things) cihazlarında güç tüketimini optimize etmek için IETF (Winter, ve diğerleri, Mart 2012) tarafından tasarlanmış bir ağ protokolüdür. IoT cihazları, sınırlı işlem gücüne, sınırlı belleğe sahip olmalarının yanında genellikle pil ile çalıştıkları için sınırlı enerjiye sahiptir. Kayıplı ağlarda enerji sorununu gidermek için tasarlanan RPL, n sayıda IoT cihazını birbiri üzerinden DAG (Directed Acyclic Graph-Yönlendirilmiş Asiklik Grafik) oluşturarak en kısa mesafeyi kurmayı amaçlar. Böylece harcanan enerjiyi optimize eder. Ancak, RPL protokolünün karmaşık altyapısı ve IoT cihazlarının düşük kapasitesi nedeniyle, bu protokol saldırılara açıktır. Bu nedenle, ağ katmanında hızlı, pratik, karmaşık olmayan ve güvenilir bir saldırı tespit sistemi geliştirmek önem arz eder. RPL ile çalışan IoT cihazlarına bir saldırı olması durumunda, 3. katman ağ paketlerinde bir anormallik meydana gelir. Bu paketlerin makine öğrenimi algoritmalarıyla işlenmesi, saldırının tespitini son derece kolaylaştıracaktır. Bu makalede (D'Hondt, Bahmad, Vanhee, & Sadre, 2015)'ın RPL protokolünde Taşma Saldırılarını, Sürüm Numarası Arttırma Saldırılarını ve Azaltılmış Rank saldırılarını simüle eden yazılımda oluşturdukları normal ve zafiyetli düğümler kullanılarak her saldırıya ait ham veri setleri elde edilmiştir. Ham veri setlerinin anlamlı hale getirilmesinin ardından, bu veri setleri 1/3 oranında test ve 2/3 oranında eğitim veri seti olarak ayırılmış, eğitim veri seti "Karar Ağacı", "Lojistik Regresyon", "Rasgele Orman", "Navie Bayes", "K En Yakın Komşu" ve "Yapay Sinir Ağları" algoritmaları ile eğitilmiş ve test edilmiştir. Saldırılara ilişkin test sonuçları karşılaştırılmış, karşılaştırma sonucunda, Taşma Saldırılarının tespitinde **%97,2** doğruluk oranı ile **Yapay Sinir Ağları** algoritması, Sürüm Numarası Artırma Saldırılarının tespitinde **%81** doğruluk oranı ile **K En Yakın Komşu** algoritması, Azaltılmış Rank saldırılarının tespitinde **%58** doğruluk oranı ile **Yapay Sinir Ağları** algoritması başarı gösterdiği tespit edilmiştir.

**Anahtar Sözcükler:** RPL, Makine Öğrenmesi, Taşma Saldırıları, Versiyon Numarası Yükseltme Saldırısı, Azaltılmış Rank Saldırısı

Bilim Kodu : 92431

Sayfa Sayısı : 84

Tez Danışmanı : Dr.Öğr.Üyesi Atınç YILMAZ

## ABSTRACT

**Comparison of Machine Learning Algorithms to Detect RPL-Based IoT Devices Vulnerability**

Murat Ugur KIRAZ

National Defense University, Hezârfen Aeronautics and Space Technologies Institute

Istanbul, January 2022

The RPL protocol (Routing Protocol for Low-Power and Lossy Networks) is a network protocol designed by IETF (Winter, et al. March 2012) to optimize power consumption in the Internet of Things (IoT) devices. IoT devices have limited processing power, limited memory, and limited energy because they usually run on batteries. Designed to solve the energy problem in lossy networks, RPL aims to establish the shortest distance by creating a DAG (Directed Acyclic Graph) of n number of IoT devices over each other. Thus, it optimizes the energy expended. However, due to the complex infrastructure of the RPL protocol and the low capacity of IoT devices, this protocol is vulnerable to attacks. Therefore, it is crucial to develop a fast, practical, uncomplicated, and reliable intrusion detection system at the network layer. An anomaly will occur in layer three network packets in the event of an attack on RPL-powered IoT devices. Processing these packets with machine learning algorithms will make it extremely easy to detect the attack. (D'Hondt, Bahmad, Vanhee, & Sadre, 2015) succeeded in simulating Flooding, Version Number Increase, and Decreased Rank Attacks. With the normal and malicious IoT motes that they created; the raw data sets of each attack were obtained. After the raw data sets were made meaningful, these data sets were separated as 1/3 test and 2/3 training data set. Datasets were trained and tested by "Decision Tree," "Logistic Regression," "Random Forest," "Naive Bayes," "K Nearest Neighbor," and "Artificial Neural Networks" algorithms. As a result of the comparison, the **Deep Learning** algorithm detected Flooding Attacks with a **97.2%** accuracy rate. The **K Nearest Neighbor** algorithm detected Version Number Increasing Attack with an **81%** accuracy rate. The **Deep Learning** algorithm detected decreased rank attacks with a **58%** accuracy rate.



**Keywords** **:** RPL, Machine Learning, Flooding Attacks, Version Number Increase Attacks, Decreased Rank Attacks

Science Code : 92431

Pages : 84

Supervisor : Assistant Professor Atinc YILMAZ

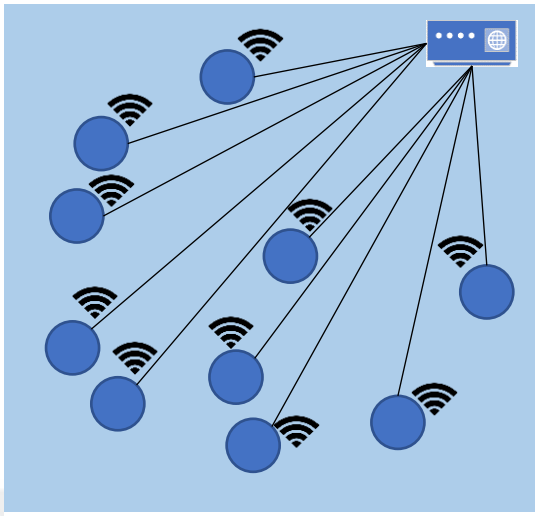
# 1. GİRİŞ

Dünya düzenini değiştiren, küreselleşmenin önünü açan ve yepyeni bir çağ başlatan iki unsur şüphesiz ki bilişim ve internettir. Bu değişimin Claude Shannon'un 1936'da henüz 21 yaşındayken yazdığı yüksek lisans tezinde, Boole cebrinin elektronik devrelerle gerçekleştirebileceğini göstermesi ile başladığını kabul edebiliriz. (Chiu, Lin, Mcferron, Petigara, & Seshasai, 2001) Modern bilgisayarın temellerinin atıldığı bu dönemden sonra, 1'ler ve 0'larla ifade edilebilen her şey insan hayatını derinden etkiledi. Her türlü veri bilgiye dönüştürüldü ve insanoğlunun muazzam kararlar almasına neden oldu.

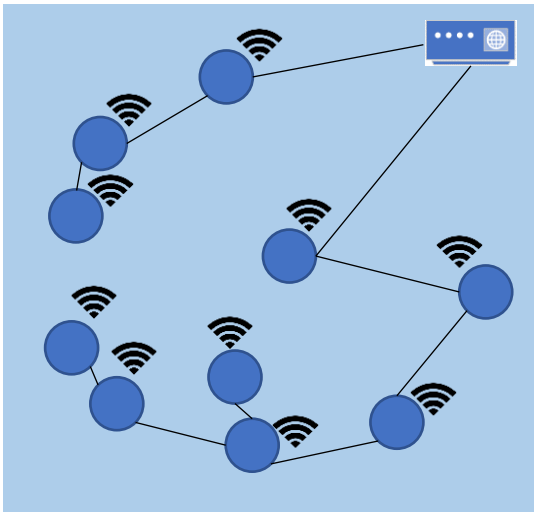
Cihazlardan, ortamdan ve hatta canlılardan elde edilen verileri toplamak, işlemek, cihazları çalıştırmak veya elde edilen bilgileri bir yere aktarmak için bazı aygıtlara ihtiyaç duyulması üzerine "Nesnelerin İnterneti" (Internet of Things-IoT) kavramı ortaya atıldı. Nesnelerin İnterneti, "belirli bir amacı gerçekleştirmek için bir ağda çalışan, veri toplayan, verileri ileten, verileri işleyen, verilerin analizini yapıp bir sonuç çıkarabilen, işlediği verilerle cihazları çalıştıran, durduran aygıtlardır." diyebiliriz. Bugün birçok otomasyon IoT cihazları sayesinde yapılabilmektedir.

İnsan hayatına girmesiyle beraber muazzam fayda sağlayan IoT cihazları, birçok noktadan veri topladığı, ürettiği veya aktardığı için, şebeke voltajı ile enerji ihtiyaçlarını doğrudan karşılamak yerine bağımsız enerji kaynaklarına ihtiyaç duymaktadırlar. Ayrıca bu cihazların çoğu kablosuz haberleşirler. Üstelik bu IoT cihazlarının çok, belki binlerce olduğu bir ortamda hepsinin bir noktaya veri transferini gerçekleştirmesi maliyet etkin bir enerji tüketimini gerçekleştirmeyecektir. Çünkü veri transfer noktasına (modem, router vs.) daha uzak olan cihaz, daha fazla güç tüketecektir. Ayrıca, uzak noktaya veri transferi esnasında veri paketlerinde kayıplar oluşacaktır. Bu sorunu çözmek için cihazların birbirleri üzerinden iletişim kurarak transfer noktasına bağlantı sağlamak büyük bir enerji tasarrufu sağlayacak ve veri paketlerindeki bozulmaları asgariye indirecektir. Tek veri transfer noktası ile iletişim sağlayan ve birbiri üzerinden iletişim kuran topoloji örneği Şekil 1.1'de gösterilmiştir.

Bu tip sistemler, çok noktadan çok noktaya (Multipoint to Multipoint) olarak adlandırılmıştır.



Tek veri noktası ile iletişim kuran topoloji



Birbiri üzerinden iletişim kuran topoloji

**Şekil 1.1: Tek Veri Noktası ile ve Birbiri Üzerinden İletişim Kuran Topoloji.**

Mart 2012 yılında İnternet Mühendisliği Görev Gücü (IETF) tarafından RPL protokolü (Düşük Güçlü ve Kayıplı Ağlar için IPv6 Yönlendirme Protokolü) ortaya çıkarılmıştır. (Winter, ve diğerleri, Mart 2012) Bu protokol ile, IoT cihazları birbirleri üzerinden veri iletişimini gerçekleştirmiş, düşük güçte çalışılması sağlanmış ve veriyi daha az kayıplı iletmiştir.

6LoWPAN protokolü için tasarlanan RPL, IoT cihazlarının güç tüketimini optimize etmeyi amaçlar. Ancak, RPL'nin iç yapısının karmaşıklığı ve 6LoWPAN cihazlarının düşük güvenlikli doğası nedeniyle bu protokol, ağ içindeki veya dışındaki saldırılara karşı savunmasızdır. Gerçi, RPL protokolü kendi bünyesinde güvenlik kontrollerini barındırır fakat bu kontrollerin uygulanmaması veya uygulanamaması, bu protokolü zafiyetlere açık hale getirecektir.

3'üncü katmanda çalışan RPL'ye yönelik saldırılar, sistemin güç tüketimini artıracağı için IoT cihazlarının normalden fazla işlem yapmasına ve veri iletmesine neden olacağı için bataryalarının erken bitmesine, dolayısıyla güç optimizasyonu için tasarlanmış olan bu sistemin, istenilen şekilde çalışmamasına neden olacaktır. Bu nedenle, **ağ katmanında, hızlı, pratik, karmaşık olmayan ve güvenilir** bir saldırı tespit sistemi geliştirmek çok önemlidir. RPL ile çalışan IoT cihazlarına bir saldırı olması durumunda, 3. katmandaki ağ paketlerinde bir anomali meydana gelecektir.

Siber güvenlik alanında, anomaliyi veya saldırıların tespit edebilmek için makine öğrenmesi metotları çok yaygın olarak kullanılmaktadır. Bilindiği üzere, makine öğrenmesi, bilgisayara eldeki verilerle matematiksel hesaplamalar yaptırılması sonucunda bilgisayarın mevcut verileri öğrenmesi, en nihayetinde eldeki veriye benzer veriler bilgisayara verildiğinde bilgisayarın kuvvetli tahmin yapabilmesidir.

OSI (Open Systems Interconnection (Açık Sistem Arabağlantısı)-3'üncü katman olan ağ katmanında çalışan RPL protokolündeki gerçekleştirilecek saldırıları, **hızlı, pratik, karmaşık olmayan ve güvenilir** yolla tespit edebilmek için, aşağıdaki aşamalar izlenildiği takdirde saldırılar tespit edilecektir.

- Nesnelerin interneti cihazlarını simüle eden bir yazılım ile normal düğümlerle (IoT cihazlarıyla) yapılacak simülasyonun veri paketlerini toplamak,
- Aynı simülasyon ile farklı saldırılar gerçekleştiren zafiyetli düğümlerle yapılacak simülasyonun veri paketlerini toplamak,
- Simülasyonlardan elde edilen verileri işlemek, normal ve zafiyetli düğümlerle yapılan verileri etiketleyerek sınıflandırmak.
- Yeni oluşturulan veri setini ile Makine Öğrenmesi algoritmaları ile eğitmek ve test etmek,
- En iyi sonucu veren makine öğrenmesi algoritmasını tespit ederek, saldırı tespit sistemini bu algoritma üzerinde kurgulamak.

Bilindiği üzere, çok farklı makine öğrenmesi algoritmaları bulunmaktadır. Bu algoritmaların birbirlerine karşı üstünlükleri ve zafiyetleri vardır.

### 1.1 Tezin Amacı

Bu yüksek lisans tezinde RPL'ye yapılan saldırıları **hızlı, pratik, karmaşık olmayan ve güvenilir** bir yolla tespit edebilmek maksadıyla OSI 3'üncü katmanı olan ağ katmanındaki veri paketleri 6 adet makine öğrenmesi algoritması ile analiz edilip aralarındaki en hızlı, etkili, karmaşık olmayan ve güvenilir algoritmayı tespit etmek amaçlanmıştır.

### 1.2 Benzer Çalışmalar

RPL protokolüne gerçekleştirilen saldırıların makine öğrenmesi metotları ile tespit edilmesinde literatürde birçok çalışma bulunmaktadır.

Yavuz, 2018 yılında gerçekleştirdiği yüksek lisans tezinde, RPL protokolü ile çalışan nesnelerin interneti cihazlarına yapılan saldırıları tespit etmek maksadıyla farklı sayıdaki normal ve zafiyetli cihazlarla Cooja simülasyonu üzerinde saldırıları gerçekleştirmiş, verileri setini elde etmiş ve elde ettiği veri seti ile derin öğrenme tabanlı bir saldırı tespit sistemi elde etmiştir. (Yavuz, 2018)

Müller ve arkadaşları 2019'da Kernel Density Estimation (KDE) ile bir makine öğrenimi yöntemi geliştirmiştir. Bu yöntem ile, Kara Delik (Black Hole), Hello Flood (Taşma saldırısı-HF) ve Sürüm Numarası Değiştirme saldırısını yüzde 84,91 gerçek pozitif (GP) ve yüzde 0,5'ten az yanlış pozitif (YP) oranı ile tespit etmeyi başarmışlardır. (Müller, Debus, Kowatsch, & Böttinger, 2019)

Neerugatti ve Reddy, çalışmalarında K-en yakın komşu algoritmasına dayanan MLTKNN adlı makine öğrenimi yaklaşımına dayalı bir saldırı tespit tekniği önermişlerdir. 30'a varan çeşitli miktarlarda IoT cihazı (düğüm) ile % 90-% 98 GP oranına ve % 0,9 - % 0,2 YP oranına ulaşmışlardır. (Neerugatti & Reddy, 2019)

Verma ve Ranga, 2019'da, RPL'ye karşı saldırıları tespit etmek için ELNIDS (Ensemble Learning based Network Intrusion Detection System- Ensemble Learning Tabanlı İzinsiz Giriş Tespit Sistemi) adlı bir Ağ Saldırısı Tespit Sistemi mimarisi tasarlamıştır. Bu tasarım Boosted Trees, Bagged Trees, Subspace Discriminant,, ve RUSBoosted Trees algoritmaları ile uygulanmıştır. Çalışmada RPL-NIDDS17 adlı veri setinin 20 özelliği kullanılarak Sinkhole, Blackhole, Sybil, Clone ID, Selective Forwarding, HF ve Local Repair saldırıları makine öğrenimi yöntemleriyle tespit edilmiştir. Boosted Trees algoritması % 94,5 ile en yüksek doğruluğu elde edilirken, Subspace Discriminant yöntemi% 77,8 ile en düşük doğruluk elde edilmiştir. (Verma & Ranga, 2019)

Belavagi ve Muniyal, 2020'de RPL protokolü ile oluşturulan sistemdeki düğüm sayısını sırası ile 10, 40 ve 100 düğüm olarak ayrı ayrı tasarlamışlardır. Her bir sistemde de zafiyetli düğümlerin oranı %10, %20 ve %30 olacak şekilde düğüm eklemişlerdir. Elde ettikleri tutarsızlık yüzdesi, enerji tüketimi, GP ve YP oranına göre şebekenin davranışını gözlemlemişlerdir. Bu çalışmada, makine öğrenimi algoritmaları için ağ paketlerinin yanı sıra diğer parametreleri de kullanmışlardır. (Belavagi & Muniyal, 2020)

Çakır, Toklu ve Yalçın, IoT ağlarında RPL'ye yönelik Taşma Saldırılarını saldırılarını tahmin etmek ve önlemek için Geçitli Tekrarlayan Birim ağ modeli tabanlı derin öğrenme algoritması önermişlerdir. Bu modeli Destek Vektör Makineleri (SVM-Support Vector Machines) ve Lojistik regresyon (LR-Logistic Regression) yöntemleriyle karşılaştırmışlardır; ayrıca, düğümlerin farklı güç durumlarını ve toplam enerji tüketimini test etmişlerdir. Sundukları model ile HF saldırılarını literatür çalışmalarına göre çok daha düşük hata oranıyla tespit etmişlerdir. (Çakır, Toklu, & Yalçın, 2020)

Shafiq ve arkadaşları, IoT güvenliğinde kullanılacak siber saldırı tespit sistemi için birçok makine öğrenimi algoritması arasından etkili bir makine öğrenimi algoritmasının seçilmesini sağlayan bir model üzerinde çalışmıştır. Bu çalışma sonucunda Naive Bayes Makine Öğrenmesi algoritmasının, IoT ağında anomalilik ve saldırı tespitini etkin bir şekilde tespit ettiği sonucuna varılmıştır. (Shafiq, Tian, Sun, Du, & Guizani, 2020) Ancak bu çalışma RPL tabanlı IoT cihazları için yapılmamıştır.

Bu tez çalışmasında, diğer çalışmalardan farklı olarak, RPL'deki Taşma Saldırıları, Sürüm Numarası Arttırma Saldırıları ve Azalan Rank saldırılarını saldırıları tespit etmede kullanılmak üzere tek bir makine öğrenimi yöntemi geliştirilmemiştir. Aksine, birden fazla makine öğrenimi yöntemini karşılaştırarak en iyi sonucu elde etmeye odaklanılmıştır. Saldırıları tespit etmek maksadıyla oluşturulan veri setinde sadece 3. katman ağ paketlerinden elde edilen veriler kullanılmıştır. Bunun nedeni oldukça basittir. Her IoT cihazı için anlık güç ve enerji tüketimi gibi parametrelerin alınması, işlenmesi ve iletilmesi, fazladan işlem ve kapasite gücü gerektirecektir. Bu nedenle çok hızlı bir şekilde elde edilebilen 3'üncü katman ağ paketleri bu çalışmada kullanılmıştır. 3'üncü katmanda iletilen normal ve zafiyetli RPL verileri 1 saniyelik çerçevelere bölünmüş, bu çerçevede yer alan paketlerin süreleri, boyutları, RPL mesaj türleri ve paketlerin oranları özetlenerek bir veri seti hazırlanmıştır. Bu veri setleri Karar Ağaçları, Lojistik Regresyon, Rastgele Orman, K En Yakın Komşu, Navie Bayes ve Yapay Sinir Ağı makine öğrenmesi algoritmaları ile test edilmiştir. Son olarak, en iyi sonucu bulmak için bu makine öğrenimi algoritmaları karşılaştırılmıştır.

Bu çalışmada, ikinci bölümde, IoT cihazları, bu cihazlarda kullanılan diğer protokoller, RPL protokolü ve RPL protokolünde gerçekleştirilen saldırılar detaylıca açıklanmıştır. Üçüncü bölümde, saldırılara ait veri setini elde etmek için deneylerin

nasıl yapıldığı detaylıca açıklanmış ve hangi yöntemlerin belirleneceği belirtilmiştir. Dördüncü bölümde ise deneyler sonucunda elde edilen veriler yorumlanmıştır.

Makine öğrenmesi algoritmalarından en çok başarı sağlayan ve en karmaşık problemleri bile kolayca tespit edebilen makine öğrenmesi algoritması elbette ki yapay sinir ağlarıdır. Yapay sinir ağları temelde nümerik verilerden lineer olmayan haritalama özelliği ve paralel çalışma özelliği ile (Yılmaz, 2015) karşımıza güçlü bir algoritma olarak çıkar. Yalnız, yapay sinir ağlarının eğitim süreleri uzun olduğu için diğerlerine göre biraz daha maliyetlidir. RPL protokolüne yapılan saldırıların tespitinde yapay sinir ağları yüksek doğrulukla sonuç vereceği beklenmektedir. Ancak, Karar Ağaçları, Lojistik Regresyon, Rastgele orman, K En Yakın Komşu, Navie Bayes gibi diğer makine öğrenmesi algoritmalarının eğitim sürelerinin daha kısa olacağı ve en az yapay sinir ağları kadar yüksek doğrulukla saldırıyı tespit edecekleri değerlendirilmektedir. (Yılmaz, 2015)

# 2. GENEL BİLGİLER

Nesnelerin İnterneti cihazları, belirli bir işlevi yerine getirmek için programlanmış ve internet veya diğer ağlar üzerinden veri transferini sağlayabilen sensorlar, aktüatörler, cihazlar veya makineler gibi donanım parçalarıdır. IoT cihazları, karar vermeyi gerçekleştirmek için bir ağ üzerinden birbirine bağlanan, sistem olmayan cihazlar topluluğu olarak da adlandırılabilir. (Lueth, 2014)

İnsan hayatının refahını artırmak maksadıyla, makine ve nesneler arasında iletişimi sağlayan, hesaplamalar yapan ve her biri ile koordinasyonu sağlayan IoT cihazlarının sayısı her geçen gün artmaktadır. 2012 yılında, 8,7 milyon cihaz varken, 2020 yılında bu rakam 50,1 milyona yükselmiştir. (Burhan, Rehman, Khan, & Kim, 2018)

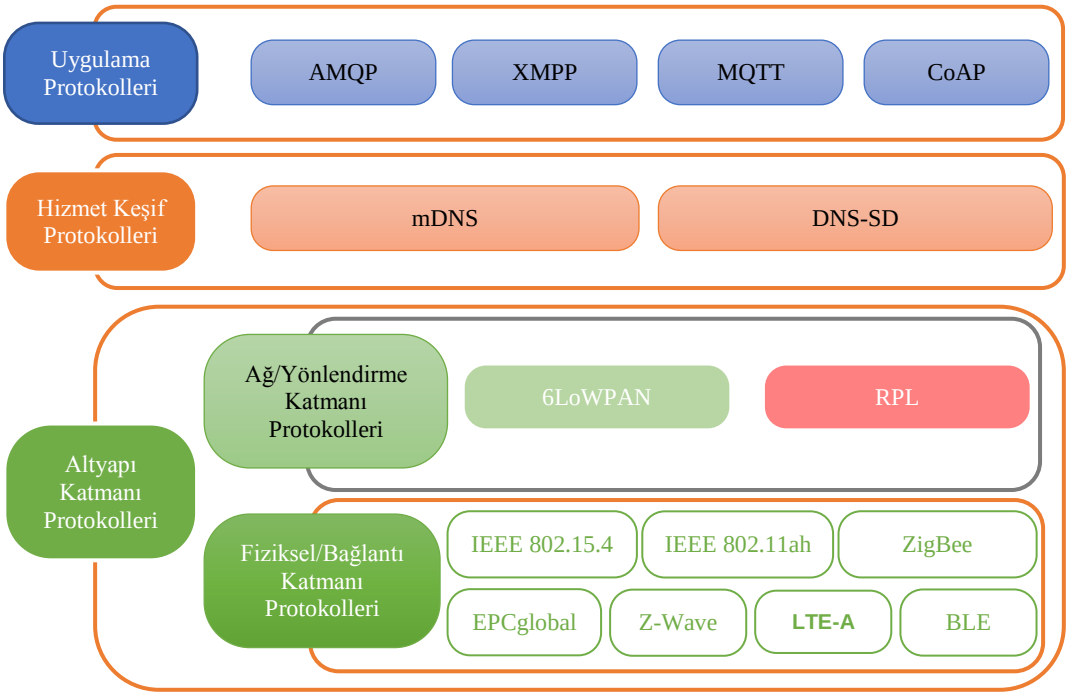
IoT cihazlarından beklenti maliyet etkinliktir. Çünkü, çok sayıda kullanılacak olan IoT cihazlarından basit işlemler yapması ve işlediği veriyi aktarması istenir. Bu isteği karşılamak için IoT cihazlarının düşük bellek kapasiteleri, düşük işlemci hızları, düşük ağ iletişim hızı bulunmaktadır. Bu nedenle, mevcut iletişim protokolleri IoT cihazları için maliyet-etkin olmayabilmektedir. Sonuç olarak IoT Cihazları için yeni iletişim protokolleri oluşturulma ihtiyacı doğmuştur.

## 2.1 IoT Protokol Yığını

Iot cihazları için, protokol yığınlarını birçok kaynakta çok farklı biçimlerde tanımlanmıştır. Ancak, genelleme yapılacak ve cihazların ortak özellikleri ele alınacak olursa, 3 protokol bloğundan söz etmek yerinde olacaktır. Bu protokoller ve katmanlar Şekil 2.1'de gösterilmiştir. Aşağıda bu protokollere ilişkin özet bilgiler bulunmaktadır.

### 2.1.1 Altyapı Katmanı Protokolleri

Bu katmandaki IoT protokolleri, farklı alt katmanlara, yani Fiziksel Katman Protokolleri, Bağlantı Katmanı Protokolleri ve Ağ / Yönlendirme Katmanı Protokolleri (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015) olarak kategorize edilebilir.




**Şekil 2.1: IoT Protokol Katmanları.**

### 2.1.1.1 EPCglobal

IoT cihazlarında RFID (Radio-Irequency IDentification- Radyo frekansı tanımlama) teknolojisi, nesneleri tanımlamak için günümüzde kullanılmaktadır. RFID etiket okuyucu, bir nesnede bulunan benzersiz etiket numarasını okur ve nesne adı hizmet uygulamalarına iletir. RFID etiketindeki benzersiz etiket numarası Elektronik Ürün Kodu (EPC-Electronic Product Code) numarası olarak bilinir. Bu EPC numarası esas olarak akıllı öğelerin tanımlanmasına ve izlenmesine yardımcı olur. EPCglobal organizasyonu, EPC ve RFID standartlarının geliştirilmesinden ve yönetilmesinden sorumludur. (Grasso, 2004)

### 2.1.1.2 Z-wave

Z - Wave, 30 m'ye kadar noktadan noktaya iletişimi desteklemek için Z - Wave Alliance ve Sigma Designs tarafından geliştirilmiş düşük güçlü bir iletişim protokolüdür. (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015)

#### 2.1.1.3 Bluetooth Düşük Enerji (BLE)

Kablosuz iletişimde, enerjiyi maliyet etkin kullanmak maksadıyla, bluetooth çekirdeğinin geliştirilerek oluşturulmuş BLE (Bluetooth Low Energy- Bluetooth Düşük Enerji) standardı, 100 metre mesafede IoT cihazları tarafından kullanılmaktadır. (Mackensen & Lai, 2012)

#### 2.1.1.4 IEEE 802.15.4

IEEE (Institute of Electrical and Electronics Engineers- Elektrik ve Elektronik Mühendisleri Enstitüsü) 802.15.4, fiziksel katman ve MAC (Media Access Control-Medya Erişim Kontrolü) katmanının işlevlerini sağlayan, LR-WPAN (Low Rate Wireless Personal Area Network-Düşük Hızlı Kablosuz Kişisel Alan Ağı) ağı için teknik bir standarttır. (Howitt & Gutierrez, 2003) IEEE 802.15.4, düşük güç tüketimi ile güvenilir yüksek veri hızlı iletişim sağlar. (Zheng & Lee, 2006) Büyük ölçekli düğüm işleme yeteneği ve düşük veri iletim güç tüketimi nedeniyle, çeşitli IoT sistemlerinde yaygın olarak kullanılmaktadır. Benzer şekilde, 802.15.4, IoT sistemlerinde iletişimi desteklemek için kullanılan farklı protokollerin geliştirilmesinin temelini oluşturur. Bu protokoller, düşük güçlü kablosuz kişisel alan ağlarında (6LoWPAN) ZigBee, IPv6'dır. 802.15.4'ün fiziksel katmanı Doğrudan Sıralı Yaygın Spektrum'a (Direct Sequence Spread Spectrum) (DSSS) dayanır ve verileri üç frekans bandında (yani 2,4 GHz, 915 MHz ve 868 MHz) iletebilir. Frekans bandına göre farklı veri hızlarında (40 ila 250 kbps) iletişimi destekler. Düşük frekanslar uzun bir mesafeyi kapsar ve daha iyi hassasiyet sağlamada iyidir. Öte yandan, yüksek frekanslar, düşük gecikmeyi ve yüksek verimliliği destekler. IEEE 802.1.4'ün MAC katmanı, ağdaki birden fazla düğümün sürekli eşzamanlı iletiminin çakışmasını azaltabilen CSMA / CA protokolüne dayanmaktadır. (Gutierrez, ve diğerleri, 2001)

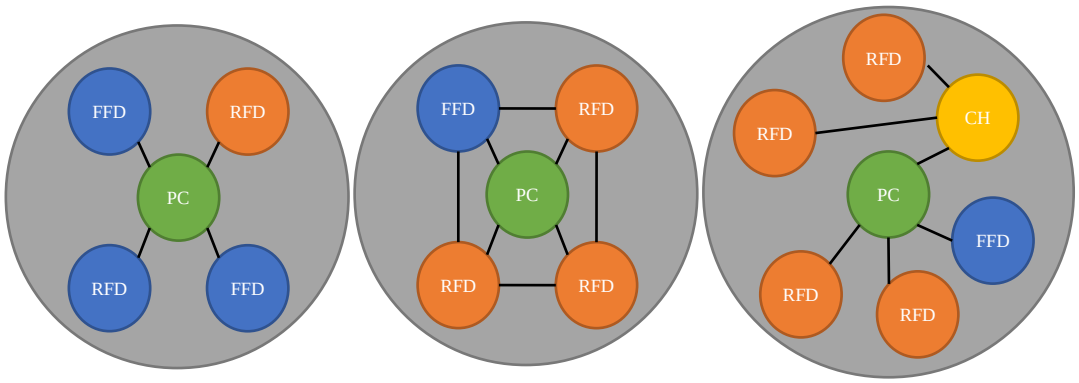
IEEE 802.15.4, iki tür ağ düğümünü desteklemek için uygulanmıştır. Bunlar, azaltılmış fonksiyonlu cihazlar (RFD-Reduced Functioned Devices) ve tam fonksiyonlu cihazlardır (FFD-Full Functioned Devices). RFD, sınırlı kaynaklara ve iletişim gereksinimlerine sahip basit bir ağ cihazıdır. Öte yandan, FFD cihazlarının daha fazla kaynağı vardır. FFD ekipmanı şunları yapabilir:

- WPAN'da bir koordinatör olarak hareket edebilir. Koordinatör, PAN (Personal Area Network (Kişisel Alan Ağı) ağını kontrol etmek ve yönetmek için kullanılan özel bir FFD düğümü türüdür.

- Sıradan bir düğüm gibi davranır.
- MAC uygulamalarını destekler.
- Yönlendirme tablosunu depolar.

IEEE 802.15.4 standardının ağ düğümleri birbirleriyle yıldız, eşler arası ve küme-ağaç topolojilerinde iletişim kurabilir. Bu topolojiler Şekil 2.2'de gösterilmiştir.

- Yıldız topolojisinde, koordinatör görevi görmesi maksadıyla en az bir FFD düğümü eklenir. RFD düğümleri, veri paketlerini diğer düğümlere aktarmak için FFD koordinatörleri aracılığıyla iletişim kurarlar. FFD düğümleri, PAN'daki RFD düğümlerinin bağlantısını kontrol etmekten ve sürdürmekten sorumludur.
- Eşler arası topolojilerde, bir koordinatör vardır ancak diğer tüm düğümler birbirleriyle doğrudan iletişim kurabilir.
- Küme topolojisi, PAN Koordinatörü (PC), Küme Başlığı (CH) ve RFD düğümlerinden oluşan bir tür eşler arası topolojidir. Bir küme ağacı ağında, bir koordinatör bu PAN düğümlerinin bir alt kümesi için bir koordinatör olarak hareket etmesi için başka bir FFD atayabilir. (Cuomo, Della Luna, Cipollone, Todorova, & Suihko, 2008)



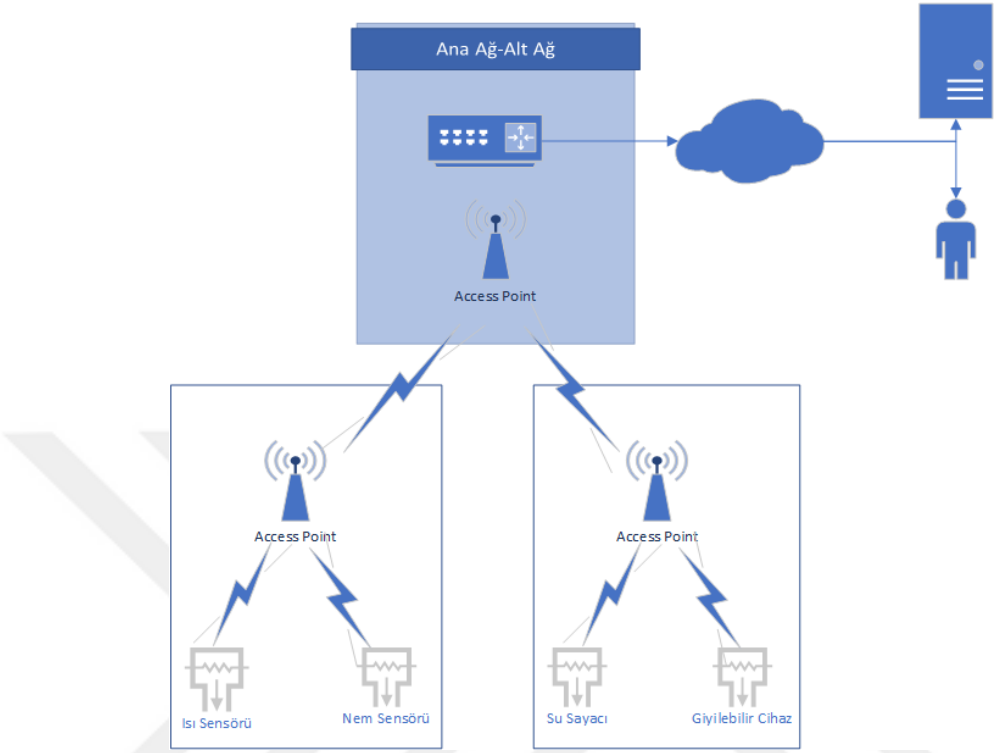

a.Yıldız Topolojisi b.Eşler Arası Topoloji c.Küme Topolojisi

**Şekil 2.2: IEEE 802.15.04 Topolojileri.**

### 2.1.1.5 IEEE 802.11ah

IEEE 802.15.4 düşük güç tüketimi sağlamaktadır ve birkaç IoT uygulaması için uygundur ancak 802.15.4 protokolü, daha geniş alanlara dağılmış çok sayıda IoT cihazından oluşan IoT uygulamaları için uygun değildir. (Olyaei, Pirskanen, Raeesi, Hazmi, & Valkama, 2013) Bu boşluğu doldurmak maksadıyla, IEEE 802.11 standardı

düşük güçlü sensör iletişim teknolojilerine altyapı oluşturması maksadı ile evrimleştirilmiş ve IEEE 802.11ah protokolü ortaya çıkmıştır. IEEE 802.11ah'ye ait örnek bir büyük ölçekli IoT sisteminin örneği Şekil-2.3'de verilmiştir.




**Şekil 2.3: IEEE 802.11ah Tabanlı IoT sistemi.**

### 2.1.1.6 ZigBee

Zigbee, kablosuz kontrol ve izleme uygulamalarında pille çalışan cihazları hedefleyen düşük maliyetli, düşük güçlü, kablosuz bir ağ standardıdır. IEEE'nin 802.15.4 protokolü üzerinde 2,4 GHz bandında çalışan ve kendi kendini onaran bir örgü ağ kullanılarak oluşturulmuştur. (Safaric & Malaric, 2006)

### 2.1.1.7 Gelişmiş Uzun Vadeli Evrim (LTE-A)

LTE–A, (Long-term Evolution – Advanced Düşük Hızlı Kablosuz Kişisel Alan Ağı Gelişmiş Uzun Vadeli Evrim) 4G LTE olarak da adlandırılır. Yüksek hızlı hücresel ağlar için bir iletişim standardıdır.

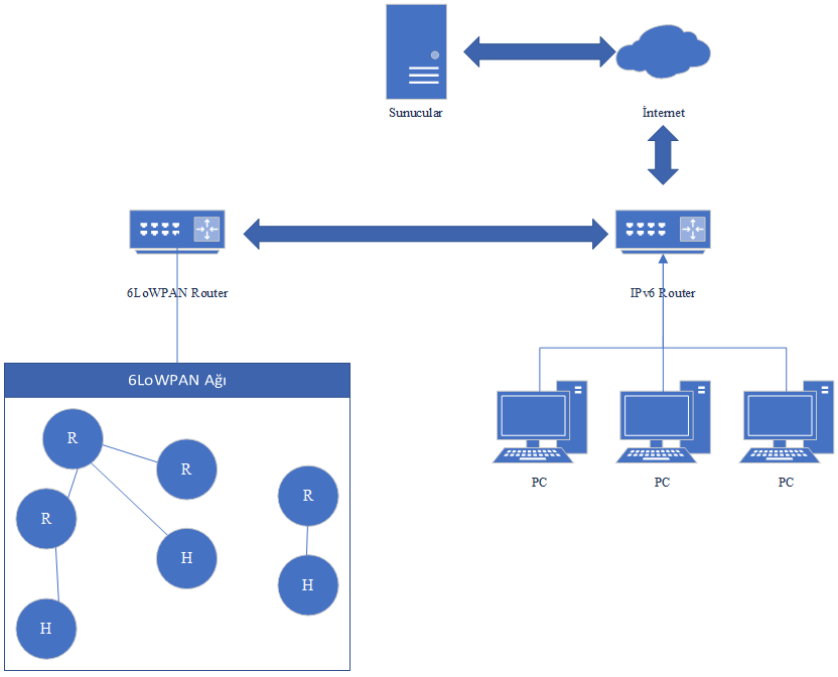
### 2.1.1.8 6LoWPAN Protokolü

6LoWPAN, 802.15.4 cihazlarla iletişimi desteklemek için IETF tarafından oluşturulan açık bir standarttır. Düşük güçlü kablosuz kişisel alan ağı üzerinden IPv6 protokolü ile

çalışır. Bu protokol BLE, düşük güçlü RD, düşük güçlü Wi-Fi, vb. bir dizi WPAN (Wireless Personal Area Network- Kablosuz Kişisel Alan Ağı) cihazını destekler.

WPAN cihazlarının değişken adres uzunlukları, sınırlı paket boyutu ve düşük bant genişliği vs., IPv6 özelliklerini barındırmak için bir uyum katmanına ihtiyaç duymaktadır. 6LoWPAN protokol uygulaması, başlık sıkıştırması ile bu problem sahasına çözüm sunar. Bu başlık sıkıştırma özelliği, çeşitli ek yükleri (yani parçalanma ek yükünü ve iletim ek yükünü) azaltır ve IoT cihazlarının IP iletişimi için tatminkâr bir destek sağlar. (Shelby & Bormann, 2009)

6LoWPAN örgü ağının mimarisi Şekil 2.4'de gösterilmiştir. 6LoWPAN ağı, yönlendiricilerden (R) ve ana bilgisayarlardan (H) oluşur. Ana bilgisayarlar, uç nokta cihazlarıdır ve verileri, nihayetinde alınan verileri hedef düğümlere yönlendiren yönlendiricilere gönderir. Hedef cihaz, ağ içindeki bir 6LoWPAN cihazı veya WPAN ağı dışındaki herhangi bir IP tabanlı cihaz olabilir. 6LoWPAN ağı, 6LoWPAN uç yönlendiricinin uygulanması yoluyla IPv6 ağına bağlanır. İnternet ağ geçidi cihazı, farklı IP tabanlı cihazların, yani PC'lerin ve sunucuların bağlandığı IPv6 yönlendiricidir. İnternette WPAN cihazları ile IP tabanlı cihazlar arasında veri alışverişi, 6LoWPAN uç yönlendiricinin uygulanmasıyla mümkündür. (Raj & Raman, 2017)




**Şekil 2.4: 6LoWPAN Ağ Mimarisi.**

#### 2.1.1.9 RPL Protokolü

Bu tez çalışmasının konusu, "RPL tabanlı nesnelerin interneti cihazlarına yapılan saldırıların tespiti için kullanılabilecek makine öğrenmesi yöntemlerinin karşılaştırılması" olduğu için RPL protokolü üzerinde biraz daha fazla durulacaktır.

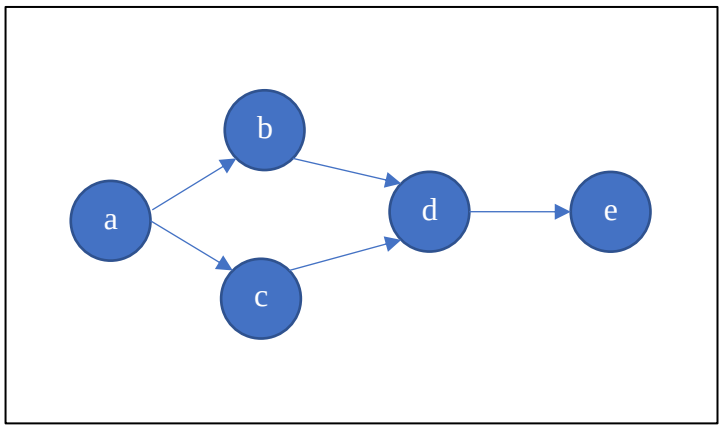
RPL, "Routing Protocol for Low-Power and Lossy Networks" yani "Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme" cümlesinin kısaltılmışıdır. RPL protokolü, IETF tarafından cihazların düşük güçlü ve kayıplı ağlarda çalışabilmesi maksadı ile Mart 2012'de oluşturulmuş bir ağ katmanı protokolüdür. (Winter, ve diğerleri, Mart 2012)

IoT cihazları doğası gereği sınırlı işlem gücüne, belleğe ve genellikle pille çalıştıkları için sınırlı enerjiye sahiptirler. Bu yüzden bu cihazların maliyet etkin bir şekilde birbirlerine bağlanmaları ve sahip oldukları kısıtlı kaynakları en etkin biçimde kullanılması kaçınılmazdır. Daha fazlası, ortamda, birden çok IoT cihazı varsa, bu cihazların kısıtlı kaynakları etkin bir biçimde kullanabilmeleri için birbirlerine noktadan noktaya (point to point) değil, birçok durumda noktadan çok noktaya (point to multipoint) veya çok noktadan noktaya (multi point to point) bağlanarak modellenmeleri gerekmektedir. Ayrıca bu ağlar binlerce düğüm içerebilir. RPL protokolünün çıkış noktası bu düşünceden doğar.

RPL protokolünün anlaşılması için, terimlerin anlaşılması gereklidir.

##### 2.1.1.9.1Terimler

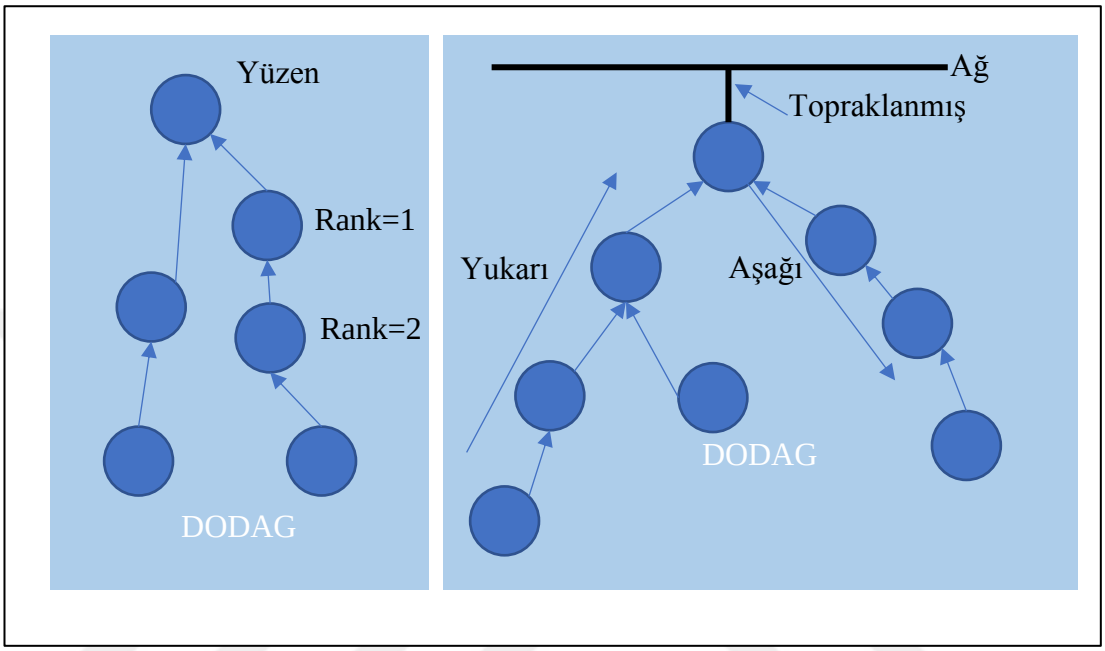
**DAG (Directed Acyclic Graph-Yönlendirilmiş Asiklik Grafik):** Yönlendirilmiş asiklik grafik. Kapalı bir döngü oluşturmayacak şekilde n adet düğümün birbirine yönlendirilmesidir. Örnek bir DAG Şekil 2.5'te gösterilmiştir.




**Şekil 2.5: Örnek Bir DAG.**

**DAG Kökü (DAG Root):** DAG'daki düğümlerin hedefidir. Bir diğer düğüme gitmez. Şikil 2.5'deki örnekte "e" düğümü DAG köküdür.

**DODAG (Destination Oriented DAG-Hedefe Yönlendirilmiş DAG):** Her düğümün tek bir hedefe ulaşmak istediği özel bir çeşit DAG türüdür. DODAG ve diğer tanımlar Şekil 2.6'da görülmektedir.




**Şekil 2.6: DODAG ve Diğer Terimler.**

**DODAG Kökü (DODAG Root):** Oluşan her bir DODAG'ın düğümlerinin bağlandığı köktür.

**Sanal DODAG kökü:** Bir Sanal DODAG kökü, iki veya daha fazla RPL yönlendiricinin sonucudur, Örneğin, 6LoWPAN Sınır Yönlendiricileri (routers), düşük güçlü ve kayıplı ağlarda DODAG durumunu senkronize etmek ve tek bir DODAG kökü gibi (birden çok arayüzle) birlikte hareket etmek için koordinasyon kurarlar.

**Yukarı:** Köke doğru yönlendirilen herhangi bir kenardır.

**Aşağı:** Kökten uzağa yönlendirilen herhangi bir kenardır.

**Rank:**Bir düğümün DODAG köküne olan uzaklığıdır. Rank aşağı doğru artar ve yukarı yönlü düşer. Rank, DODAG'ın Hedef Fonksiyonuna bağlı olarak hesaplanır.

**Hedef Fonksiyonu:** Bir düğümün köke ne kadar uzaklıkta olduğunu belirleyen fonksiyondur. Burada uzaklık kavramı programcı veya tasarımcı tarafından belirlenir.

Burada “uzaklık” kavrama en aza indirgememiz gereken parametredir. Bu parametre enerji veya gecikme olabilir.

**Amaç Kod Noktası (OCP):** (Objective Code Point) Bir OCP, DODAG'ın hangi Hedef Fonksiyonunu kullandığını belirten bir tanımlayıcıdır.

**RPL Olay Kimliği:** Bir RPL olay kimliği (RPL Instance ID), bir ağ içindeki benzersiz bir tanımlayıcıdır. Aynı RPL olay kimliğine sahip DODAG'lar aynı hedef fonksiyonunu paylaşır.

**RPL Örneği:** Bir RPL Örneği, bir RPL olay kimliği paylaşan bir veya daha fazla DODAG kümesidir. En fazla, bir RPL düğümü, bir RPL Örneğinde bir DODAG'a ait olabilir. Her RPL olayı, diğer RPL olaylarından bağımsız olarak çalışır.

**DODAGID:** DODAG kimliği, bir DODAG kökünün tanımlayıcısıdır. DODAGID, LLN'deki (Low-Power and Lossy Networks- Düşük Güçlü ve Kayıplı Ağlar) bir RPL Örneği kapsamında benzersizdir. Her DODAG'ın bir IPv6 Kimliği (128 bit) vardır. Bu kimlik yalnızca DODAG köküne verilir ve kök kimliği değiştirmediği sürece de değişmez.

**DODAG Sürümü:** Bir DODAG Sürümü, belirli bir DODAGID ile bir DODAG'ın belirli bir yinelemesidir. DODAG'ın her yeni şekli, yeni bir sürüm anlamına gelir.

**DODAG Sürüm Numarası:** DODAG Sürüm Numarası, bir DODAG'ın yeni bir Sürümünü oluşturmak için kök tarafından artırılan sıralı bir sayaçtır. Bir DODAG Sürümü benzersiz bir şekilde (RPL Olay Kimliğinde, DODAG kimliğinde, DODAG Sürüm Numarasında) başlıkla tanımlanır.

**Hedef:** Hedef, Bir DODAG'ın ulaşmak istediği yerdir. Kablolu veya kablosuz bir ağ olabilir. Hedef, Amaç işlevinden farklıdır. Amaç işlevinde amacımız en aza indirmektir. Ancak Hedef, gitmek istediğimiz yerdir.

**Topraklanmış (Grounded):** DODAG hedefine ulaştığında topraklanmıştır.

**Yüzen (Floating):** DODAG, topraklanmamışsa yüzer. Yüzen bir DODAG'ın hedefi karşılamak için gerekli özelliklere sahip olması beklenmez. Bununla birlikte, DODAG içindeki diğer düğümlere bağlantı sağlayabilir.

**DODAG ebeveyni (DODAG Parent):** Ebeveyn, okların işaret ettiği yerdir. Bir çocuk ise, okun geldiği yerdir. Ebeveynlerin birden fazla çocuğu olabilir, Benzer şekilde bir çocuğun birden fazla ebeveyni olabilir.

**Alt DODAG (Sub-DODAG):** Belirli bir DODAG'ın alt ağacıdır.

**Yerel DODAG (Local DODAG):** Yerel DODAG'lar yalnızca bir kök düğümü içerir ve bu tek kök düğümün, yerel bir RPL Olay Kimliği ile tanımlanan bir RPL Örneğini diğer düğümlerle koordinasyon olmadan tahsis etmesine ve yönetmesine izin verir. Tipik olarak bu, LLN içindeki bir varış noktasına giden rotaları optimize etmek için yapılır.

**Evrensel DODAG (Global DODAG):** Evrensel DODAG, diğer birçok düğüm arasında koordine edilebilen RPL Olay Kimliğine sahip DODAG'tır.

**Saklanan (Stored):** Saklanan düğümleri bütün yönlendirme tablosu bilgisini kaydeder. Bu düğümler sadece kendi ebeveyn düğümlerini bilirler.

**Saklanmayan (Non-Stored):** Basittirler, herhangi bir yönlendirme tablosunu saklamazlar, sadece ebeveynlerini bilirler.

Aşağıda düğümler arasında gönderilen mesajlara ait terimler yer almaktadır.

**DIO:(DODAG Information Object- DODAG Bilgi Nesnesi): Bu** mesaj, aşağı doğru çok noktaya yayınlanır. Bir DODAG'daki belirli bir düğüm, bu mesajı çok noktaya yayınlayarak diğer düğümlerin topraklanmış olup olmadığını veya saklanan olup olmadığı bilgisini iletir.

**DIS (DODAG Information Solicitation-DODAG Bilgi İsteme):** Duyuru duyulmadığında ve bir düğüm bir DODAG'a katılmak isterse bir kontrol mesajı gönderir, Bunun için herhangi bir DODAG olup olmadığını bilmek ister.

**DAO (DODAG Advertisement Object-DODAG Duyuru Nesnesi):** Bir çocuk düğüm tarafından ebeveyn düğüme veya köke gönderilen bir istektir. Bu mesaj ile, bir düğüm, çocuk düğüm olarak bir DODAG'a katılmak için izin ister.

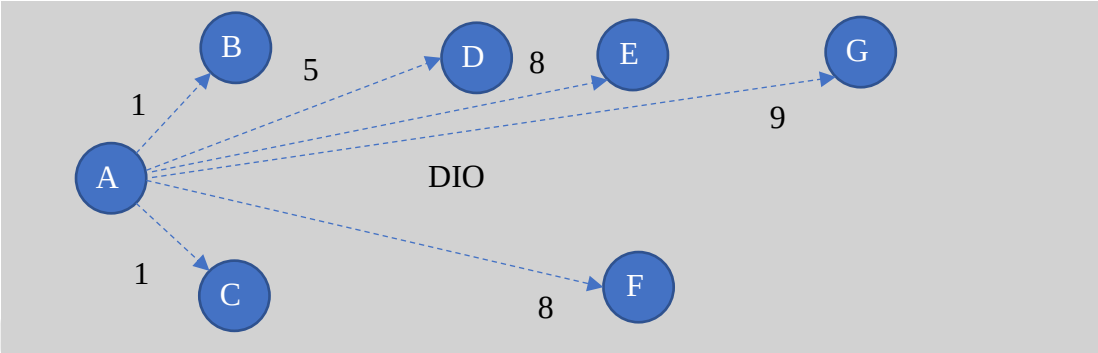
**DAO-ACK (DODAG Advertisement Object Acknowledgement- DODAG Duyuru Nesnesi Kabulü):** Bir kök veya ebeveyn tarafından çocuğa gönderilen bir yanıttır, bu yanıt Evet veya Hayır olabilir. (Winter, ve diğerleri, Mart 2012)

#### 2.1.1.9.2 RPL Protokolünün İşleyişi

Kök, DODAG'da özel bir düğümdür. Tüm düğümler DODAG'da kök olma özelliğine sahip değillerdir.

A, B, C, D, E, F, G düğümlerinden oluşan bir DODAG'ımız olsun. A düğümü ise kök düğümü olsun. Diğer düğümlerin uzaklıkları sırasıyla B=1, C=1, D=5, E=8, F=8, G=9 olsun. Bu durumda, aşağıdaki adımlar olacaktır.

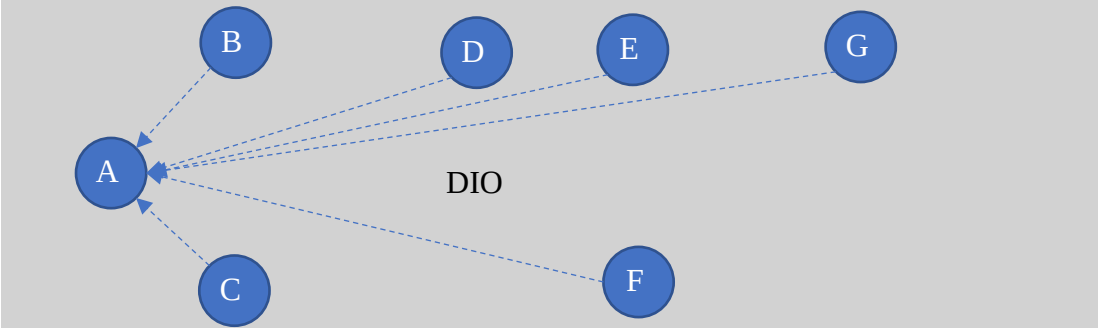
1. Öncelikle kök düğümü olarak belirlenen A Düğümü Şekil 2.7'de görüldüğü gibi tüm düğümlere DIO mesajı yayınlar.




**Şekil 2.7: DIO Mesajının Yayınlanması.**

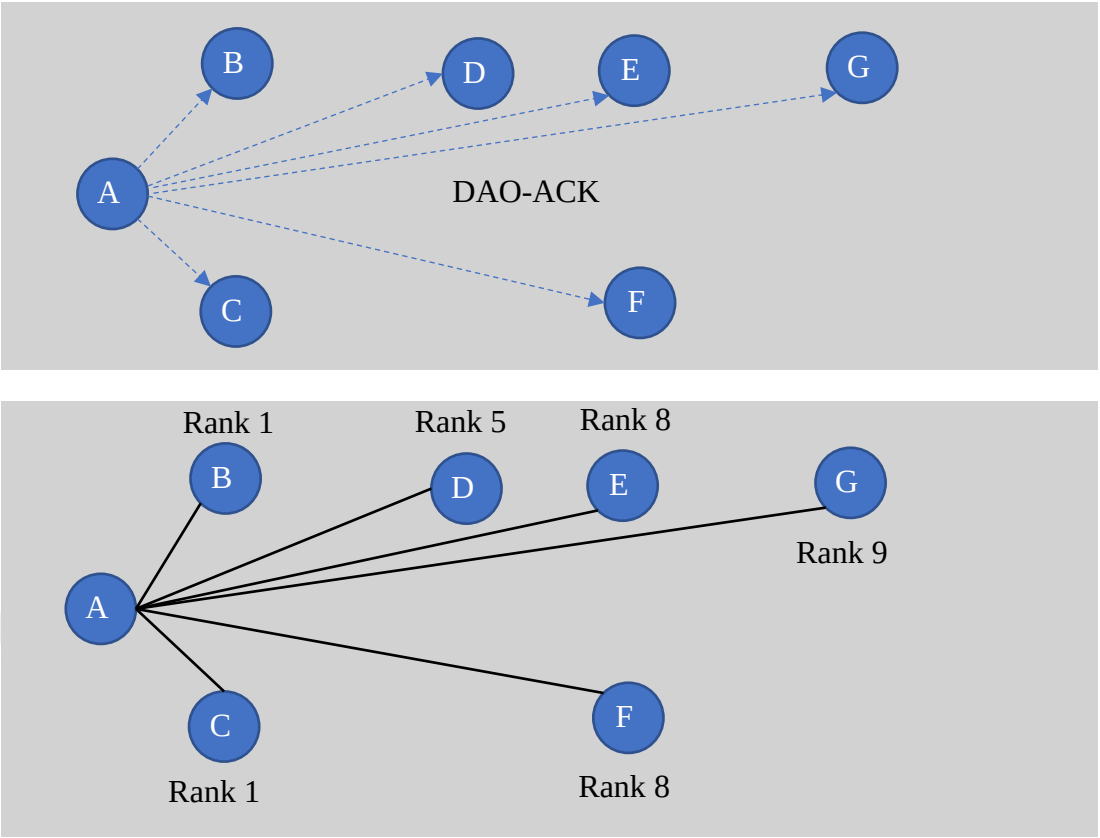
2. B,C,D,E,F,G düğümleri DIO'ları aldıktan sonra DODAG oluşturmaya başlayacaktır. Bu düğümler, DIO mesajı ile birlikte, A'dan mesafelerinin Sırasıyla 1,1,3,4,4,6 olduğunu da öğrenirler.

3. Sonrasında, B,C,D,E,F,G düğümleri A'ya DAO mesajını şekil 2.8'deki gibi iletirler.



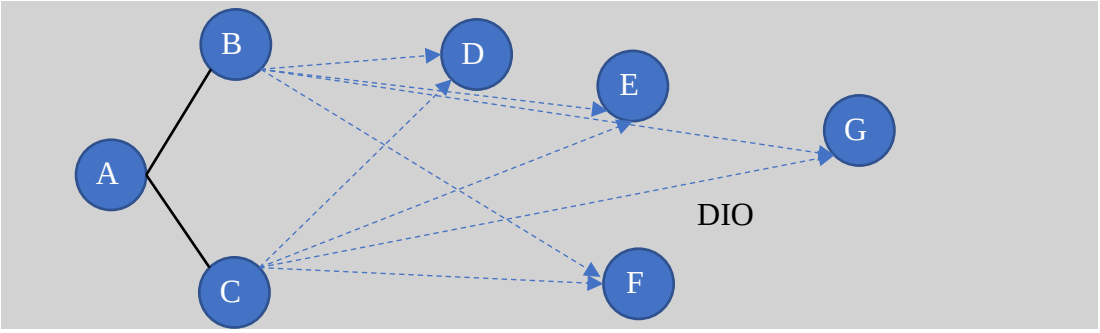

**Şekil 2.8: DAO Mesajının Yayınlanması.**

4. A, DAO-ACK mesajını tüm düğümlere gönderir ve tüm düğümleri kabul eder. Böylece şekil 2.9'daki gibi bir DODAG oluşur.




**Şekil 2.9: DAO-ACK Mesajı ve DODAG Oluşması.**

5. Bu aşamadan sonra, rankı en düşük olan düğümler, kök gibi davranır ve yukarıdaki süreçler tekrarlanır. B ve C diğer düğümlere Şekil 2.10'da görüldüğü gibi DIO mesajı gönderir ve böylelikle kendilerine olan düğüm uzaklıkları da belirlenir.

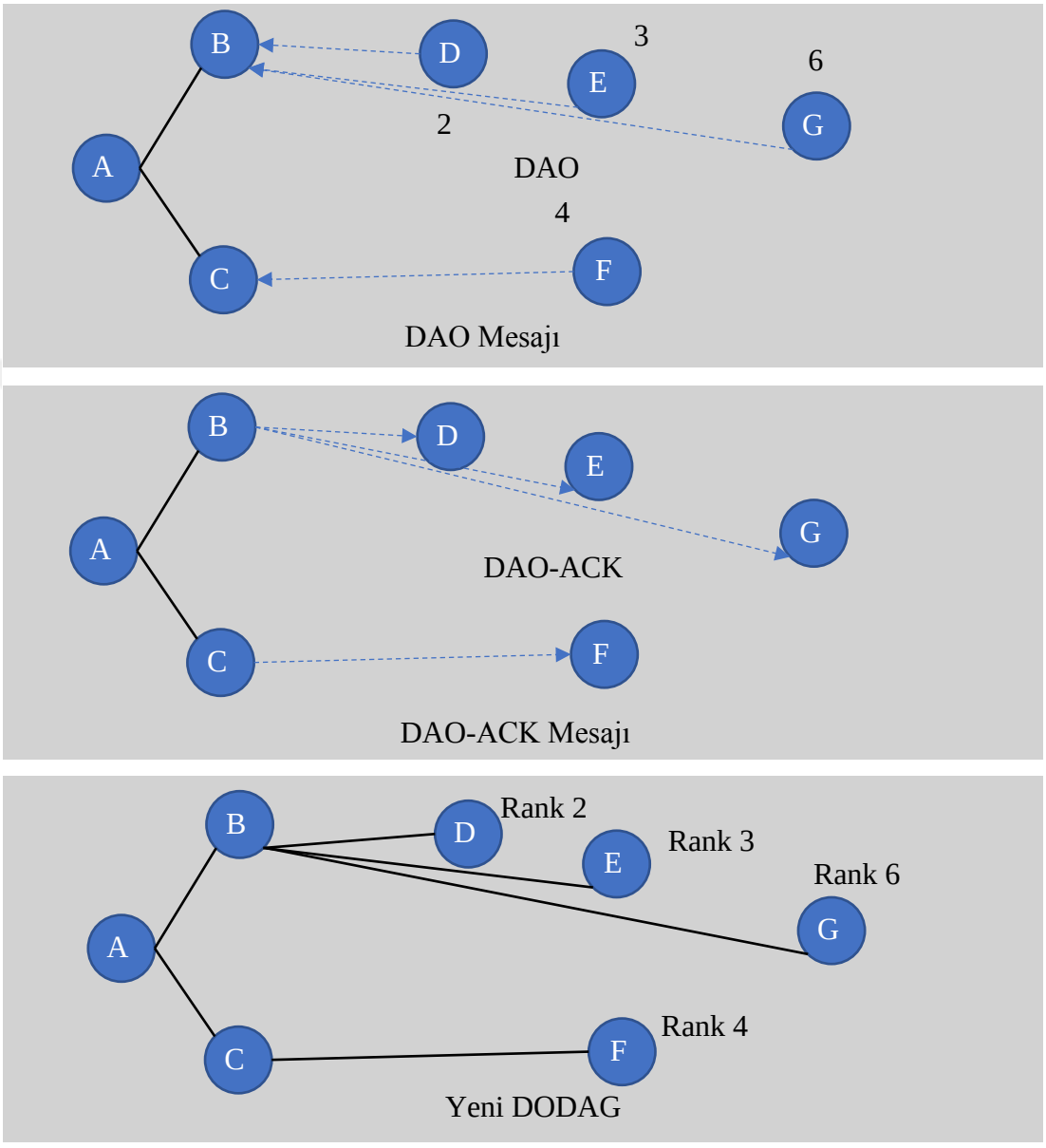



**Şekil 2.10: Yeni düğümlerin DIO Mesajı Göndermesi.**

Yeni durumda, B düğümüne D,E,F,G düğümlerinin mesafesi sırası ile 2,3,5,6 olsun. C düğümüne de D,E,F,G düğümlerinin mesafesi sırası ile 3,5,4,7 olsun.

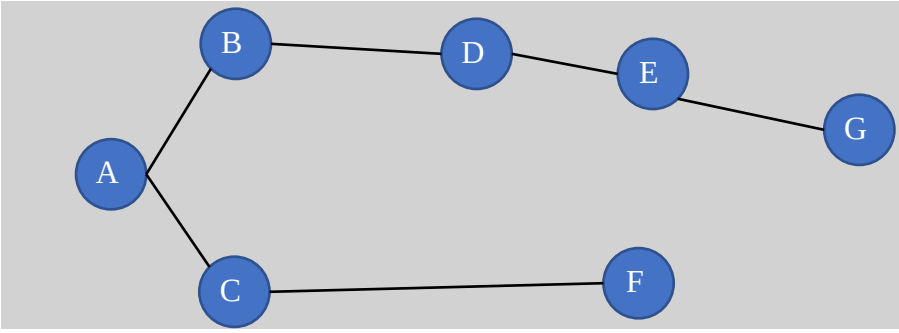
6. D,E,F,G düğümleri DIO mesajlarını aldığında DIO mesajlarını gönderen B ve C düğümlerinin kendilerine olan mesafesini anlar. D,E,G düğümleri B düğümüne C düğümünden daha yakın olduğu için B düğümüne DAO mesajı gönderir. F düğümü,

C düğümüne B düğümünden daha yakın olduğu için C düğümüne DAO mesajı gönderir. Böylelikle B ve C düğümleri DAO-ACK mesajı gönderir ve yeni DODAG oluşur. Bu süreç döngü Şekil 2.11'de görüldüğü gibi DODAG tamamlanana kadar devam eder.



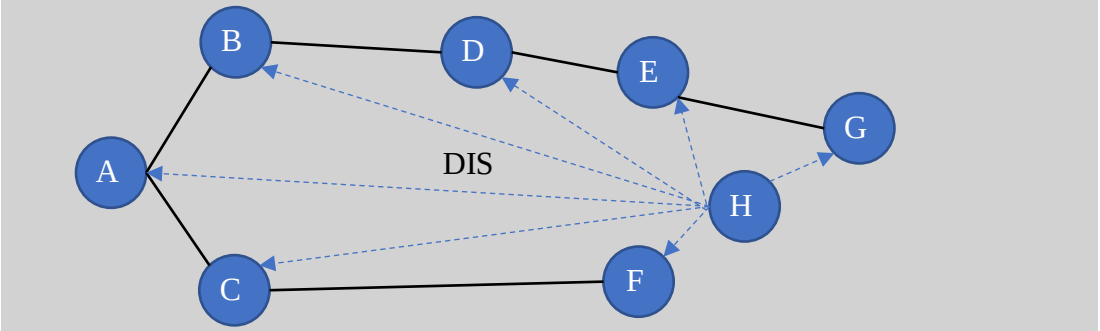

**Şekil 2.11: Yeni DODAG Oluşumu.**

7. Bir sonraki adımda en düşük ranka sahip olan D düğümü DIO mesajı gönderecek ve döngü en maliyet etlkin yol bulununcaya kadar devam edecektir. Son durum Şekil 2.12'de gösterilmiştir.

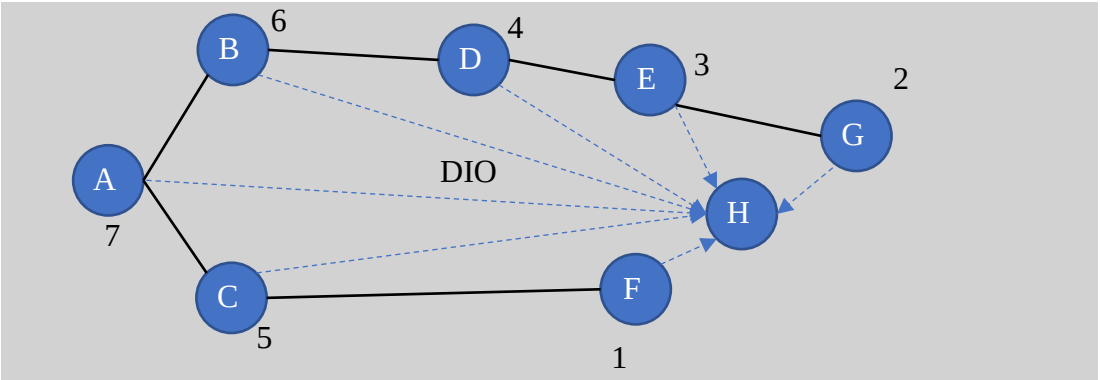



**Şekil 2.12: DODAG'ın tamamlanması.**

8. Eğer yeni bir düğüm, DODAG'a katılmak isterse, bu düğüm DODAG'a DIS mesajı gönderir. DODAG'ın bu mesajı almasından sonra, DIO mesajı gönderilir, mesafeler hesaplanır, Yeni düğüm DAO mesajı gönderir, DODAG yeni düğümü DAO-ACK mesajı ile kabul eder veya reddeder. Yeni düğüm tarafından DIS mesajının gönderilmesi Şekil 2.13'de ve DAO, DAO-ACK mesajlarının gönderilerek yeni DODAG'ın oluşturulması Şekil2.14'de gösterilmiştir. (Winter, ve diğerleri, Mart 2012)

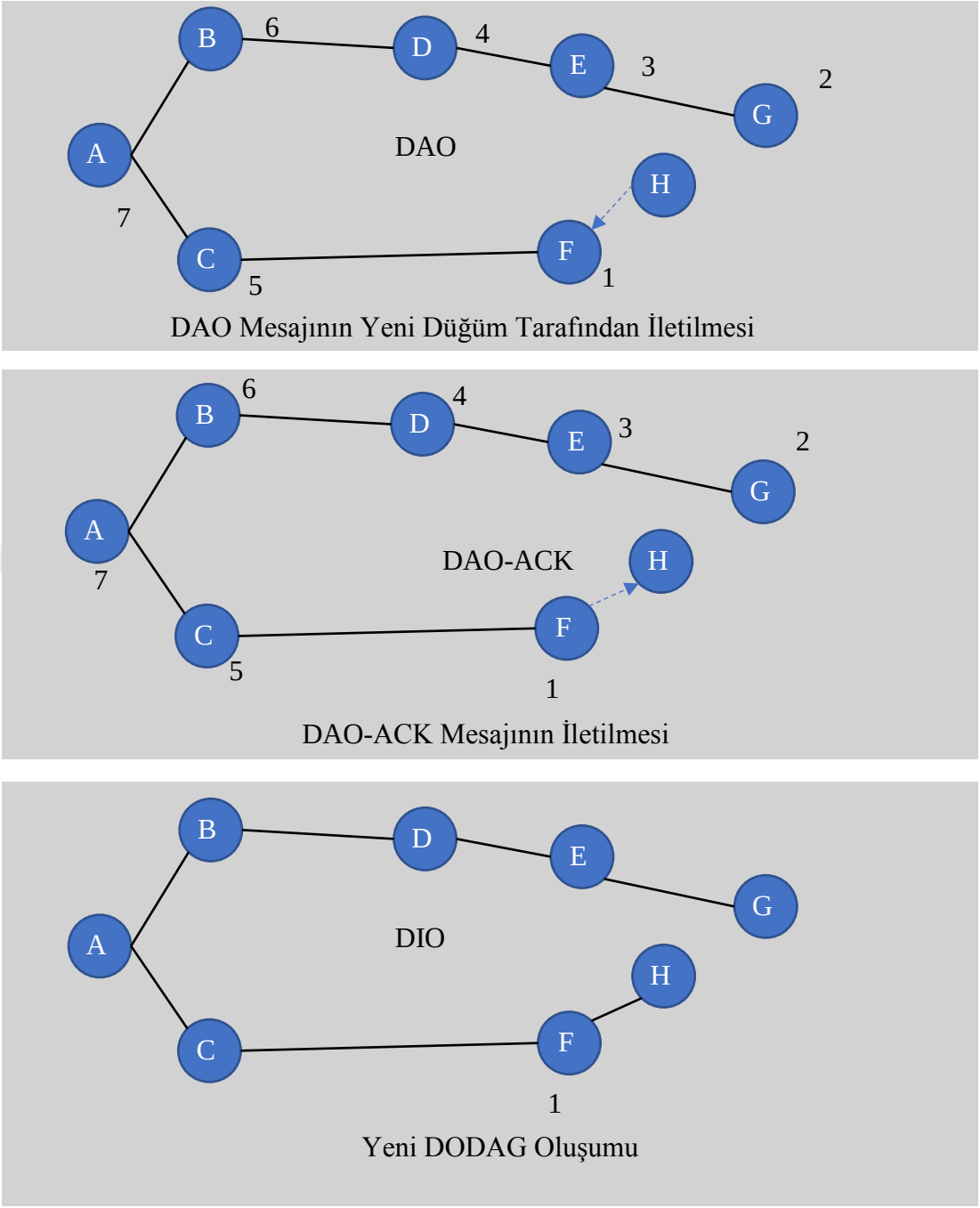



DIS Mesajının Yeni Düğüm Tarafından İletilmesi




DIO Mesajının DODAG Tarafından İletilmesi

**Şekil 2.13: DIS ve DIO Mesajlarının İletilmesi.**




**Şekil 2.14: DAO, DAO-ACK Mesajları ve Yeni DODAG Oluşumu.**

#### 2.1.1.9.3 RPL Protokolü İle İlgili Diğer Bilgiler

DODAG'ın göze çarpan özelliği olarak, yalnızca tek bir kök düğüme sahip olduğu, diğer düğümlerin, üst düğüm hakkında bilgi içerdiği fakat hiçbir düğümün alt düğümler hakkında bilgi içermediğini söylemek mümkündür.

RPL protokolü IPv6 protokolünü kullanır.

### 2.1.2 Servis Keşif Protokolleri

Birçok elemandan oluşan IoT ağındaki kaynaklar ve hizmetlerin gerek kaydının tutulması gerekse keşfi için otomatik ve verimli bir kaynak yönetimi mekanizması oluşturulması hayati derecede önemlidir. Hizmet keşif protokolleri olarak Çok Noktaya Yayın Etki Alanı Adı Sistemi (Multicast Domain Name System) (mDNS) ve Etki Alanı Adı Sistem Hizmet Keşfi'nden (Domain Name System Service Discovery) (DNS-SD) bahsedebiliriz.

#### 2.1.2.1 Çok Noktaya Yayın Alan Adı Sistemi (mDNS)

mDNS, bir IETF standardıdır. (Cheshire & Krochmal, Multicast DNS, Şubat 2013) Yerel alan ağında, çok miktarda IoT cihazı bulunuyorsa ve her bir IoT cihazının IP adresini ezberlemek mümkün değilse, mDNS servisinin kullanılması mantıklıdır. IP adreslerine isim verilmesini sağlar. Tek noktaya yayın yapan DNS sunucusuna benzer şekilde çalışır. DNS ad alanı herhangi bir ek yapılandırma olmadan yerel olarak kullanıldığı için çok esnektir. mDNS, otomatik yapılandırma sağlaması, cihazları yönetmek için ek altyapı gerekmemesi, yüksek düzeyde hata toleransı göstermesi gibi nedenlerle yerleşik IP tabanlı cihazlar için tercih edilen bir yöntemdir.

IoT cihazlar kümesinde mDNS'in kullanılması şu şekilde olacaktır: İstek yapan IoT cihazı, kimlikleri sorgulamak için önce yerel etki alanındaki tüm düğümlere IP mesajı gönderir. Bu mesajla, bir istemci belirli bir ada sahip farklı cihazlardan yanıt vermesini ister. İstekler alındıktan sonra, adını içeren hedef cihaz, IP adresini içeren bir çok noktaya yayın yanıt paketi gönderir. IoT ağındaki tüm cihazlar, bu hizmeti daha sonra kullanmak için hedef cihazın adını ve ilgili IP adresini önbellekte depolar.

#### 2.1.2.2 DNS Hizmet Keşfi (DNS-SD)

DNS-SD protokolü de bir IETF standardıdır. (Cheshire & Krochmal, DNS-Based Service Discovery, Şubat 2013) IoT sistemlerinde, DNS-SD kullanıcıların istenen hizmetleri keşfetmesini kolaylaştırmak için standart DNS mesajlarını kullanır. DNS-SD protokolü, ek yapılandırma olmadan cihazların bağlanmasına yardımcı olur. DNS-SD'de hizmet keşfi, iki aşamalı bir süreçtir,

- Gerekli hizmetlerin ana bilgisayar adlarının bulunması
- IP adreslerinin mDNS kullanılarak ana bilgisayar adları ile eşleştirilmesi.

### 2.1.3 Uygulama Katmanı Protokolleri

IoT cihazları ile ilgili birçok uygulama katmanı protokolü geliştirilmiştir. Aşağıda sık kullanılan IoT uygulama katmanı protokollerinden CoAP (Constrained Application Protocol- Kısıtlı Uygulama Protokolü), DDS (The Data Distribution Service- Veri Dağıtım Hizmeti), MQTT (Message Queuing Telemetry Transport- Mesaj Kuyruğu Telemetri Aktarımı), AMQP (Advanced Message Queuing Protocol- Gelişmiş Mesaj Kuyruk Protokolü) ve XMPP (Extensible Messaging and Presence Protocol- Genişletilebilir Mesajlaşma ve Durum Protokolü) protokolleri açıklanacaktır.

#### 2.1.3.1 Kısıtlı Uygulama Protokolü (CoAP)

Kısıtlı Uygulama Protokolü (CoAP), IETF tarafından hazırlanmıştır. CoAP, bir IP ağında çalışan kaynak kısıtlı cihazlar için Temsili Durum Aktarımına (REST) dayalı bir web aktarım protokolüdür. CoAP, HTTP ile birçok benzerliğe sahiptir, ancak aynı zamanda bazı temel farklılıklara sahiptir. Kısıtlı cihazlar sayı olarak büyük olabilir, ancak genellikle işlev veya konum bakımından birbirleriyle ilişkilidir. Örneğin, bir binadaki tüm ışık anahtarları bir gruba ait olabilir ve tüm termostatlar başka bir gruba ait olabilir. Gruplar konuşlandırılmadan önce önceden konfigüre edilebilir veya operasyon sırasında dinamik olarak oluşturulabilir. Bilginin bir grup aygıta gönderilmesi veya bu aygıttan alınması gerekiyorsa, grup iletişim mekanizmaları iletişimin verimliliğini ve gecikmesini artırabilir ve belirli bir uygulama için bant genişliği gereksinimlerini azaltabilir. HTTP, CoAP grup iletişimine eşdeğer herhangi bir işlevi desteklemez. (Rahman & Dijk, 2014)

#### 2.1.3.2 Veri Dağıtım Hizmeti (DDS)

Veri Dağıtım Hizmeti (DDS), dağıtık mimarideki cihazlar için oluşturulmuş bir uygulama katmanı protokolüdür. Bunu sağlamak için abonelere bilgi sağlamaktan sorumlu olan DCPS (Merkezi Veri Yayınlama-Abone Olma) protokolünün kullanabileceği standartlaştırılmış bir yazılım uygulama programlama arabirimi (API) sunar. DDS, bir "altyapı" çözümü olarak uygulandığından, herhangi bir yazılım uygulaması için iletişim ara yüzü olarak eklenebilir. (Pardo- Castellote, Farabaugh , & Warren, 2005)

#### 2.1.3.3 Mesaj Kuyruğu Telemetri Aktarımı (MQTT)

MQTT, OASIS tarafından hazırlanan bir protokoldür. (Banks & Gupta, 28 Eylül 2014) MQTT, bir İstemci Sunucusu yayınlama / abone olma mesajlaşma aktarım

protokolüdür. Küçük bir kod izinin gerekli olduğu ve / veya ağ bant genişliğinin önemli olduğu Makineden Makineye (M2M) ve Nesnelerin İnterneti (IoT) bağlamları gibi kısıtlı ortamlar dahil olmak üzere birçok durumda kullanım için tasarlanmıştır.

#### 2.1.3.4 Gelişmiş Mesaj Kuyruk Protokolü (AMQP)

AMQP ve OASIS tarafından hazırlanan bir protokoldür. AMQP, açık standart olan, mesaj odaklı, güvenilir ve güvenli iletişim sağlayan, en çok bir kez, en az bir kez ve tam olarak bir kez teslimatı destekleyen hem noktadan noktaya hem de yayınlama abonelik modelleri için uygun olan ve bir uygulama katmanı protokolüdür. (OASIS Advanced Message Queuing Protocol (AMQP), 2012)

#### 2.1.3.5 Genişletilebilir Mesajlaşma ve Durum Protokolü (XMPP)

Genişletilebilir Mesajlaşma ve Durum Protokolü (XMPP) XML (eXtensible Markup Language- Genişletilebilir İşaretleme Dili) tabanlı, açık, güvenli, spam içermeyen bir uygulama katmanı protokolüdür. (P. Saint, 2004) Aşağıdaki hususları sağlar:

- Kullanıcıların platformdan bağımsız anlık mesajlaşma (yani çok taraflı sohbet, sesli ve görüntülü arama) aracılığıyla birbirleriyle iletişim kurmasına olanak tanır,
- Anlık mesajlaşma uygulamalarının kimlik doğrulama, şifreleme, erişim kontrolü ve gizlilik gerçekleştirmesini sağlar,
- IoT sisteminde dosya aktarımına, oyun oynamaya ve sosyal ağ hizmetlerine izin verir.

## 2.2 RPL Protokolünde Uygulanan Saldırılar

Tezin 2.1.1.9 maddesinde RPL protokolü ile ilgili detaylı bilgi verilmiştir. Bu bölümde, RPL protokolünde uygulanan saldırılara değinilecektir.

6LoWPAN protokolü için tasarlanmış olan RPL, çok sayıda ve karmaşık yapıda olan IoT cihazlarının birbirleri ile olan iletişimlerinde güç sarfiyatının optimize edilmesini amaçlar. Enerjinin optimum seviyede kullanılması maliyet-etkinlik açısından önemli olduğu için, RPL protokolünün kusursuz işlemesi hayati önem taşır. Ancak, RPL protokolünün kendi içerisindeki karmaşıklığı ve aynı zamanda 6LoWPAN cihazlarının düşük güvenlik doğası ağın içerisinden veya dışarısından oluşabilecek

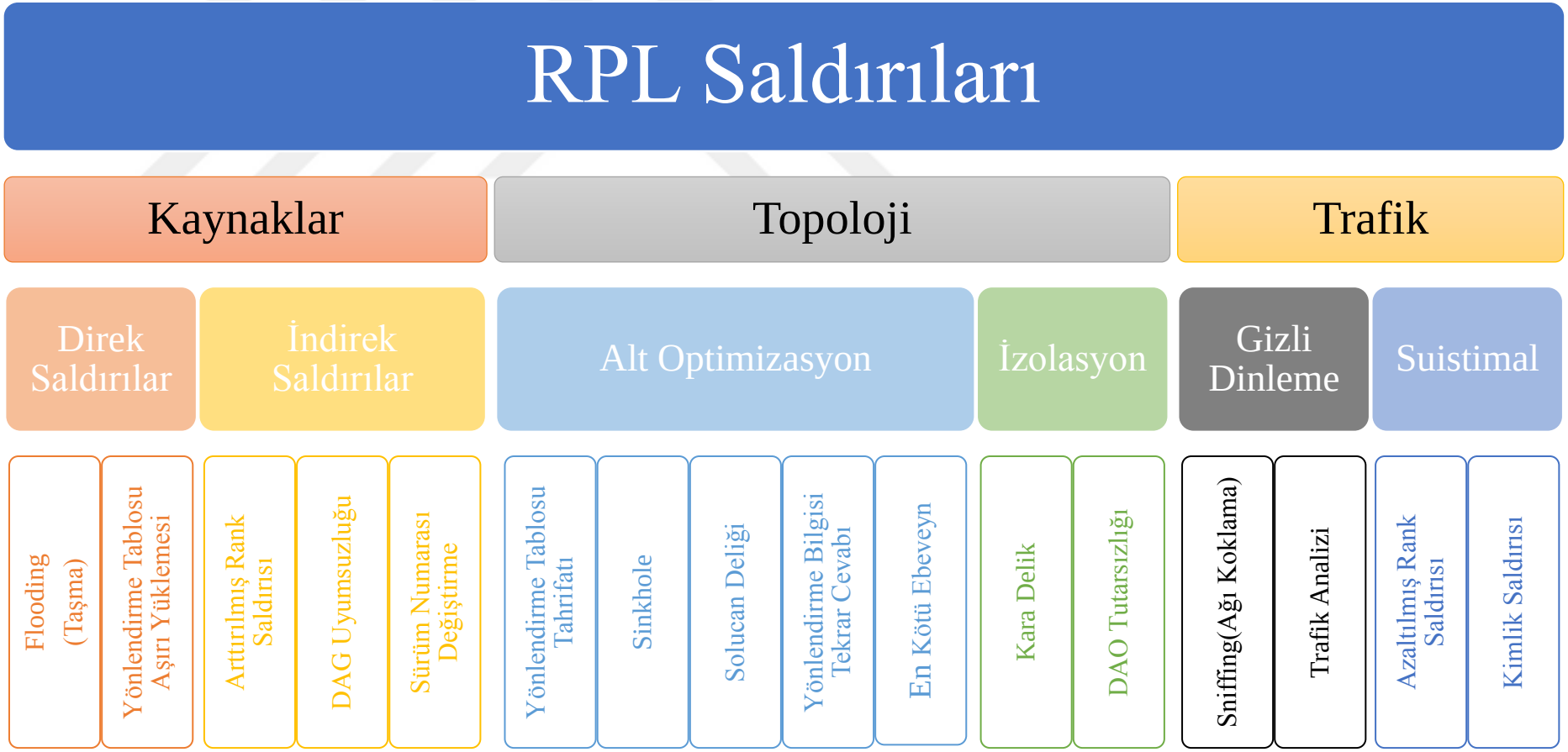
saldırılara karşı zafiyet barındırır. (Le, Loo, Luo, & Lasebae, 2011)Sonuçta DODAG yapısında bulunan herhangi bir düğümdeki zafiyet tüm sistemi etkileyecektir.

RPL protokolü kendi içerisinde elbette bir güvenlik sistemini barındırmaktadır. Bunlar, aşağıdaki gibi sıralanabilir:

- Güvenli Olmayan Mod: Bu güvenlik modunda, RPL, Güvenlik bölümleri olmayan temel DIS, DIO, DAO ve DAO-ACK mesajlarını kullanır. Bir ağ, bağlantı katmanı güvenliği gibi diğer güvenlik mekanizmalarını kullanıyor olabileceğinden, güvenli olmayan mod, tüm mesajların herhangi bir koruma olmadan gönderildiği anlamına gelmez.
- Önceden yüklenmiş Mod: Bu güvenlik modunda, RPL güvenli mesajlar kullanır. Bir RPL Örneğine katılmak için bir düğümün önceden yüklenmiş bir anahtara sahip olması gerekir. Düğümler bunu mesaj gizliliği, bütünlüğü ve özgünlüğü sağlamak için kullanır. Bir düğüm, bu önceden yüklenmiş anahtarı kullanarak RPL ağına bir ana bilgisayar veya bir yönlendirici olarak katılabilir.
- Kimlik Doğrulama Modu: Bu güvenlik modunda, önceden yüklenmiş mod'a ek olarak, düğümün ağa yönlendirici olarak katılması için, bir düğümün bir anahtar otoritesinden ikinci bir anahtar alması gerekir. Bu anahtar yetkili, ikinci anahtarı sağlamadan önce, istemcinin yönlendirici olmasına izin verildiğini doğrulayabilir. Kimliği doğrulanmış mod, simetrik algoritmalar tarafından desteklenemez. Bu spesifikasyonun yazılmasından itibaren, RPL yalnızca simetrik algoritmaları destekler: kimliği doğrulanmış mod, gelecekteki potansiyel kriptografik ilkellerin yararı için dahil edilmiştir. (Winter, ve diğerleri, Mart 2012)

Yukarıda sıralanan güvenlik tedbirlerinden "güvenli olmayan mod" kullanıldığında veya diğer doğrulama modlarının güvenlik politikaları atlatıldığında saldırılar gerçekleşecektir.

(Mayzaud, Badonnel, & Chrisment, 2016), RPL protokolü ile ilgili yapılan saldırılarla ilgili yazdıkları makalede saldırıları Şekil 2.15'de gösterildiği gibi sınıflamışlardır.




**Şekil 2.15: RPL Saldırıları.**

### 2.2.1 Kaynaklara Karşı Yapılan Saldırılar

Kaynaklara yönelik saldırılarda, DODAG'daki zafiyetli düğüm veya düğümler vasıtası ile normal düğümlerin kaynaklarını tüketmek için gereksiz işlemler gerçekleştirilmesi sağlanır. Bu saldırı ile düğümdeki enerji veya bellek tüketilir, veya işlemci fuzuli çalıştırılır. Bu tip saldırılar ile ağ ömrü istenilenden çok kısa sürede tükenir. (Mayzaud, Badonnel, & Chrisment, 2016)

#### 2.2.1.1 Direk Saldırılar

Doğrudan saldırılarda, zafiyetli düğüm kaynakların tükenmesinden doğrudan sorumludur. Bu, tipik olarak, depolama modu etkin olduğunda, taşma (flooding) saldırıları gerçekleştirerek veya yönlendirme tablolarına göre aşırı yükleme saldırıları gerçekleştirerek yapılabilir. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.1.1.1 Yönlendirme Tablosu Aşırı Yüklemesi Saldırısı

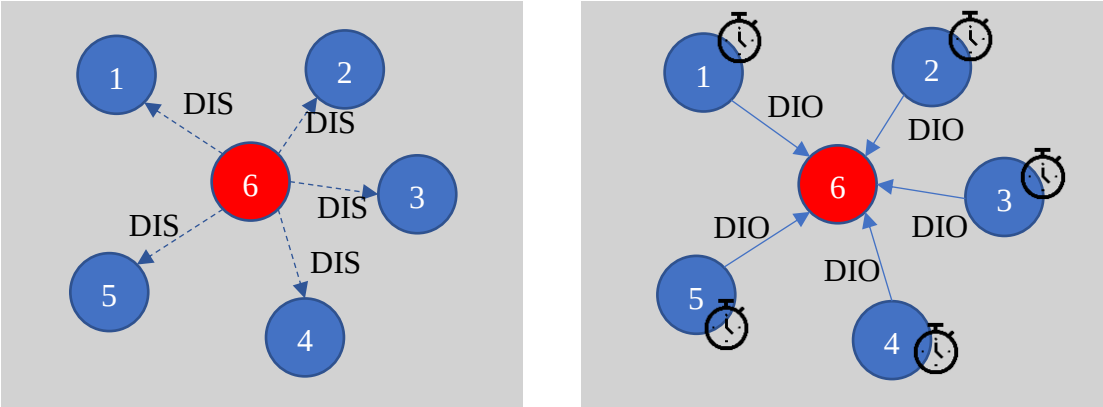
Bu saldırı, hedeflenen düğümün yönlendirme tablosunu dolduran DAO mesajlarını kullanarak sahte yollar duyurulması ile yapılır. Bu doygunluk, yeni normal yolların oluşturulmasını engeller ve ağın işleyişini etkiler. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.1.1.2 Taşma Saldırıları(Flooding Attacks)

Taşma saldırıları, bir ağda büyük miktarda trafik oluşturmayı ve düğümleri ve bağlantıları kullanılamaz hale getirmeyi içerir. Bu saldırılarda, tüm ağ düğümlerinin veya bir kısmının kaynakları tükenebilir. Bu tip mesajlara HELLO-Flood saldırısı da denilebilir. Bu saldırılar iki şekilde yapılabilir:

- Saldırgan bir düğüm ağın tamamına sürekli DIS mesajları gönderir. Mesajı alan düğümlerin damlama zamanlayıcıları (Trickle timer) sıfırlanır (reset) Bu durum Şekil-2.16'da tasvir edilmiştir.
- Saldırgan bir düğüm belirli bir düğüme DIS mesajları gönderir. Mesajı alan düğüm, DIO mesajı ile karşılık verir.

Her iki durum da, ağ tıkanıklığına ve ayrıca RPL düğümlerinin doygunluğuna yol açar. (Mayzaud, Badonnel, & Chrisment, 2016)




**Şekil 2.16: HELLO-Flood Saldırısı**

#### 2.2.1.2 Dolaylı Saldırılar

Kötü niyetli düğümün diğer düğümlerin ağ için aşırı yük oluşturmasına neden olan saldırılardır.

##### 2.2.1.2.1 Arttırılmış Rank Saldırısı

RPL ağında, her bir düğüm bir sıra değeri ile ilişkilendirilir ve kök düğüme göre grafik yapısındaki konumuna karşılık gelir. İkinci bölümde bahsedildiği gibi, DODAG'ın döngüsel olmayan yapısını korumak için düğüm sırası her zaman aşağı yönde artmaktadır. Bu yüzden bir düğümün rankı üst düğümün rankından daima büyük omak zorundadır. Bu saldırı türü, zafiyetli düğümün, olması gerekenden daha yüksek bir sıra değeri bildirmesi ile gerçekleşir. (Mayzaud, Badonnel, & Chrisment, 2016)

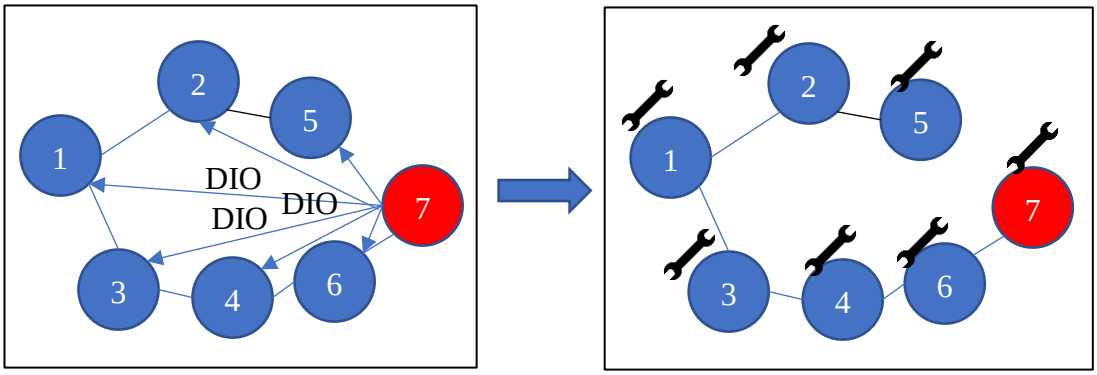
##### 2.2.1.2.2 DAG Uyumsuzluğu

Bu saldırının hemen amacı, zafiyetli düğüm ile hedeflenen düğümün DIO damlama zamanlayıcısını (Trickle timer) sıfırlamaya zorlamaktır. Bu durumda, bu düğüm DIO mesajlarını daha sık iletmeye başlar ve RPL ağında yerel istikrarsızlık oluşturur. Bu aynı zamanda düğümlerin pilini tüketir ve bağlantıların kullanılabilirliğini etkiler. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.1.2.3 Sürüm Numarası Değiştirme

DODAG sürüm numarası, bir DODAG'ın yeni bir sürümünü oluşturmak için kök tarafından artırılan sıralı bir sayaçtır. Bir DODAG Sürümü benzersiz bir şekilde (RPL Olay Kimliğinde, DODAG kimliğinde, DODAG Sürüm Numarasında) başlıkla tanımlanır. Sürüm numarası yalnızca kök tarafından artırılır. Artırıldığı zaman ise DODAG'ın yeniden yapılandırılması gerekir, buna aynı zamanda genel onarım da

denir. Daha eski bir değer, düğümün yeni DODAG grafiğine taşınmadığını ve bir üst düğüm olarak kullanılamayacağını gösterir. Zaafiyetli bir düğüm, versiyon numarasını değiştirip komşularına ilettiğinde tüm DODAG grafiği gereksiz bir biçimde yeniden oluşur. DODAG'ın art arda gereksiz yeniden yapılandırılması, mesaj ek yükünü önemli ölçüde artırır, düğüm kaynaklarını şekil 2.17'de gösterildiği gibi tüketir ve ağı tıkar. (Mayzaud, Badonnel, & Chrisment, 2016)




**Şekil 2.17: Sürüm Numarası Değiştirme Saldırısı.**

### 2.2.2 Topolojiye Yapılan Saldırılar

RPL protokolüne yönelik saldırılar, ağ topolojisini de hedefleyebilir.

#### 2.2.2.1 Alt Optimizasyon Saldırıları

Alt optimizasyon saldırıları, optimum DODAG oluşumunu önleyen saldırılardır.

##### 2.2.2.1.1 Yönlendirme Tablosu Tahrifatı

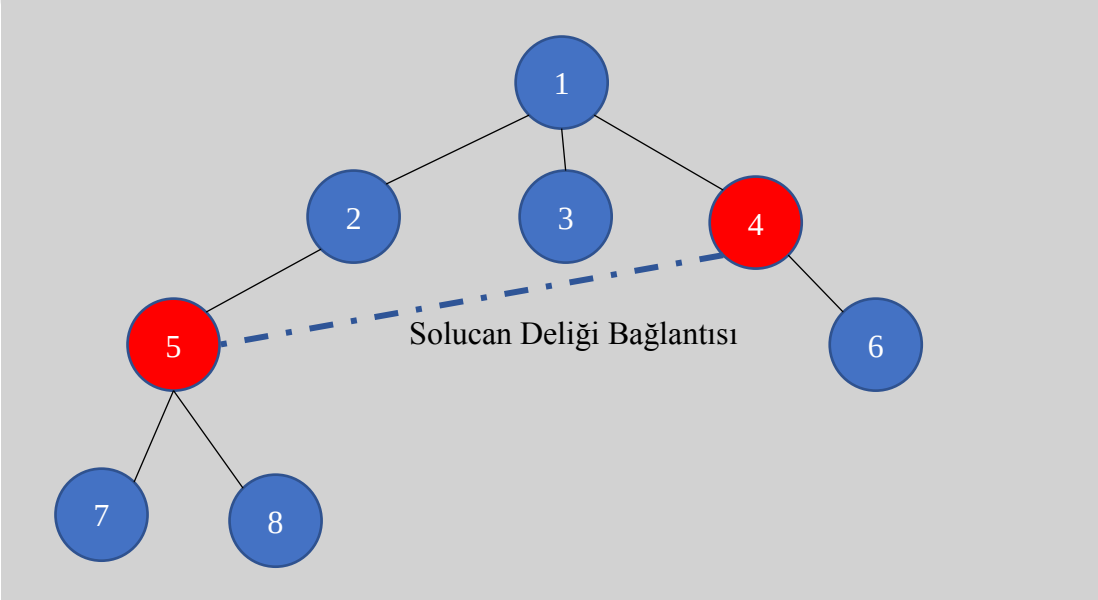
Zaafiyetli düğümün alt DODAG'da olmayan düğümlere yönelik rotaları bildirmesi ile gerçekleştirilir. Bu nedenle ağda daha uzun gecikmeye, paket düşüşlerine veya ağ tıkanıklığına neden olabilir. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.2.1.2 Sinkhole

Böyle bir saldırı iki adımda gerçekleşir. Birincisi, kötü amaçlı düğüm, sahte bilgi verilerinin reklamını yaparak (örneğin, üstün kalitede yukarı ve aşağı bağlantılar) çok fazla trafik çekmeyi başarır. Ardından, trafiği gayri meşru bir şekilde aldıktan sonra, onu değiştirir veya bırakır. (Mayzaud, Badonnel, & Chrisment, 2016)

#### 2.2.2.1.3 Solucan Deliği

Solucan deliği saldırıları, özel bir ağ bağlantısıyla birbirine bağlanan bir çift RPL saldırgan düğümünün (A ve B düğümlerinin) kullanımı olarak tanımlanabilir. Şekil 2.18'de bir örnek gösterilmektedir. Bu senaryoda, düğüm 4 tarafından alınan her paket, daha sonra yeniden oynatılmak üzere solucan deliği yoluyla düğüm 5'e iletilir. Görevler birbirinin yerine geçebilir olduğundan, düğüm 4, düğüm 5 ile aynı işlemleri gerçekleştirebilir. Kablosuz ağlar söz konusu olduğunda, bu saldırıyı gerçekleştirmek daha kolaydır. Bu saldırı, yönlendirme yolunu bozar. Bir saldırgan, yönlendirme bilgisini ağın başka bir kısmına, yani aslında uzak olan düğümlere tünellerse, birbirlerini yan yanaymış gibi görür. Sonuç olarak, amaç fonksiyonuna göre optimize edilmemiş rotalar oluşturabilirler. (Mayzaud, Badonnel, & Chrisment, 2016)




**Şekil 2.18: Solucan Deliği Saldırısı.**

#### 2.2.2.1.4 En Kötü Ebeveyn Saldırısı

Bu saldırı, zafiyetli düğümün en maliyetli ebeveyni seçmesi ve ona bağlanması ile gerçekleşir. Bu da güç tüketimini artırır ve optimal ağ yolunu bozar.

#### 2.2.2.1.5 Yönlendirme Bilgisi Tekrar Cevabı

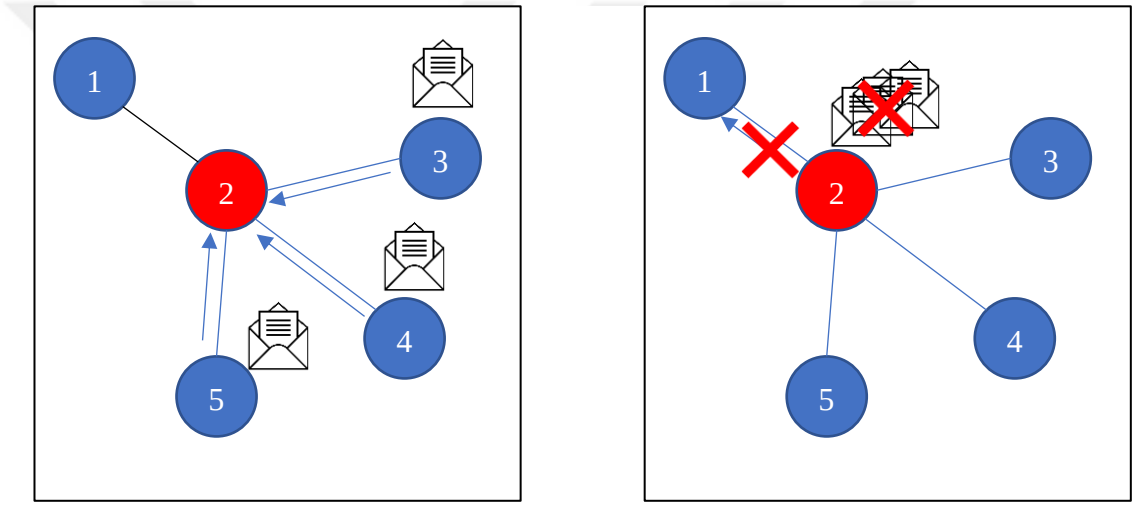
Bu saldırı, zafiyetli bir düğümün diğer düğümlerden geçerli kontrol mesajlarını kaydetmesi ve bunları daha sonra ağda iletmesi ile gerçekleşir. (Mayzaud, Badonnel, & Chrisment, 2016)

#### 2.2.2.2 İzolasyon Saldırıları

DODAG'da herhangi bir düğümün veya düğüm grubunun, diğer düğümlerle iletişim kuramamasına yönelik yapılan saldırılardır.

##### 2.2.2.2.1 Kara Delik (Black Hole) Saldırıları

Bir kara delik saldırısında, şekil 2.19'da tasvir edildiği gibi zafiyetli düğüm, iletmesi gereken tüm paketleri düşürür. Bu saldırı, trafiğin büyük bir kısmının kaybolmasına neden olan bir sinkhole saldırısıyla birleştirildiğinde çok zarar verici olabilir. Bir tür hizmet reddi (DoS) saldırısı olarak görülebilir. Saldırgan grafikte stratejik bir konumda bulunuyorsa, ağdan birkaç düğümü izole edebilir. (Mayzaud, Badonnel, & Chrisment, 2016)




**Şekil 2.19: Kara Delik (Black Hole) Saldırısı.**

##### 2.2.2.2.2 DAO Tutarsızlığı Saldırıları

DODAG'ın oluştuğu sırada, bir düğüm daha önce bir DAO mesajından öğrenilen aşağı doğru bir rotaya sahip olabilir, ancak bu rota artık alt düğümün yönlendirme tablosunda geçerli olmayabilir. Bu durumda DAO mesaj tutarsızlıklarını gidermek için RPL, DAO tutarsızlık döngüsü kurtarma adı verilen bir mekanizma sağlar. Bu saldırı, bu mekanizmanın suiistimali ile gerçekleşir. (Mayzaud, Badonnel, & Chrisment, 2016)

### 2.2.3 Trafiğe Yönelik Yapılan Saldırılar

RPL ağ trafiğini hedefleyen saldırılar ile ilgilidir.

#### 2.2.3.1 Gizli Dinleme Saldırıları

RPL ağlarının yaygın yapısı, ağ trafiğinin koklanması ve analizi gibi gizli dinleme faaliyetlerini gerçekleştiren kötü niyetli düğümlerin konuşlandırılmasını kolaylaştırabilir. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.3.1.1 Koklama (Sniffing) Saldırıları

Koklama saldırısı, ağ üzerinden iletilen paketleri dinlemekten oluşur. Bu saldırı, kablolu ve kablosuz ağlarda çok yaygındır ve iletişimin gizliliğini tehlikeye atar Bu saldırının pasif yapısı nedeniyle tespit edilmesi zordur. Koklamayı önlemenin tek yolu, şifrelemedir. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.3.1.2 Trafik Analizi

Trafik analizi, bir bağlantı üzerindeki trafiğin özelliklerini ve modellerini kullanarak yönlendirme bilgilerini almayı amaçlar. Bu saldırı, paketler şifrelenmiş olsa bile gerçekleştirilebilir. Amaç, koklama saldırıları gibi, ebeveyn / çocuk ilişkilerini tanımlayarak topolojinin kısmi bir görünümü gibi RPL ağı hakkında bilgi toplamaktır. Bu saldırı sayesinde kötü niyetli bir düğüm toplanan bilgilerle muhtemelen başka saldırılar da gerçekleştirebilir. Sonuçlar saldırganın derecesine bağlıdır. Bu, kök düğüme yakınsa, büyük miktarda trafiği işleyebilir ve bu nedenle düğümün bir alt DODAG'ın kenarında bulunduğundan daha fazla bilgi alabilir. (Mayzaud, Badonnel, & Chrisment, 2016)

#### 2.2.3.2 Suiistimal Saldırıları

Kötüye kullanma saldırılarında, meşru bir düğümün kimliği gasp edilir veya aşırı performans talep edilir. Bu saldırılar kendi başına RPL ağı için çok zararlı değil. Bununla birlikte, önceki iki ana kategoride görülenler gibi diğer saldırılar için genellikle bir ilk adım olarak kullanılırlar. Saldırganın ağı ve topolojisini daha iyi anlamasına, daha iyi erişim elde etmesine veya trafiğin büyük bir bölümünü durdurmasına olanak tanırlar. (Mayzaud, Badonnel, & Chrisment, 2016)

##### 2.2.3.2.1 Azaltılmış Rank Saldırısı

Kötü amaçlı bir düğüm, anormal bir şekilde daha düşük bir rank duyurduğunda, performansını aşar. Sonuç olarak, birçok meşru düğüm saldırgan aracılığıyla DODAG grafiğine bağlanır. (Mayzaud, Badonnel, & Chrisment, 2016)

Bu saldırı bir ağa zarar vermez, ancak diğer yapı taşlarıyla birleştirmek çok etkili olabilir çünkü saldırganın kötü niyetli düğüm üzerinden bazı trafiği tünellemesine izin verir (örneğin, gizli dinleme için).

#### 2.2.3.2.2 Kimlik Saldırısı

Bir saldırganın, kök düğümü belirlemek için ağ trafiğini koklayabilir. Bu tanımlama gerçekleştirildiğinde, DODAG kökünün adresini yanıltabilir ve ağ üzerinde kontrolü ele alabilir. (Mayzaud, Badonnel, & Chrisment, 2016)

# 3. VERİ SETİNİN OLUŞTURULMASI, MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRILMASI, DENEYLERİN YAPILMASI

Bu tez çalışmasında, RPL protokolünde gerçekleşebilecek Taşma Saldırıları, Versiyon Numarası Yükseltme Saldırıları ve Azaltılmış Rank saldırısına ilişkin deneyler yapılıp veri seti oluşturulacaktır. Bu maksatla aşağıdaki aşamalar izlenecektir:

- Her saldırı için, normal IoT cihazları (düğüm) ile simülasyon yapılması ve ağ paketi verilerinin kaydedilmesi,
- Her saldırı için, zafiyetli IoT cihazı içeren cihazlarla (düğümlerle) simülasyon yapılması ve ağ paketi verilerinin kaydedilmesi,
- Her saldırı için, elde edilen ağ paketi verilerinin anlamlı hale getirilmesi, zafiyetli düğüm içeren ağ paketleri verilerinin "1" etiketi ile, normal düğüm içeren ağ paketleri verilerinin "0" etiketi ile sınıflandırılması. Böylelikle 3 adet sınıflandırılmış veri setinin oluşturulması,
- Her saldırı için, sınıflandırılmış veri setlerinin test ve eğitim veri seti olarak ayrılması, normalizasyon işleminin yapılması,
- Her saldırı için ayrılan eğitim veri setinin 6 farklı makine öğrenmesi algoritmaları ile eğitilmesi,
- Her saldırı için ayrılan test veri setinin 6 farklı makine öğrenmesi algoritmaları ile test edilmesi, doğruluk oranlarının ve eğitim sürelerinin belirlenmesi
- Sonuçların karşılaştırılması.

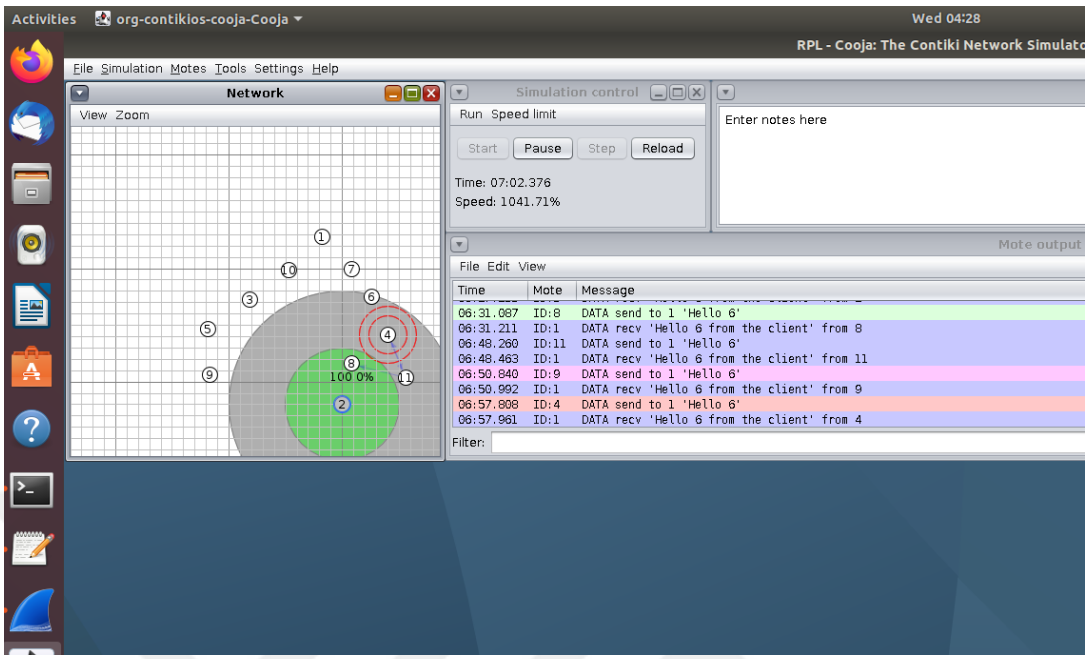
## 3.1 Saldırıların Simülasyonu

### 3.1.1 Contiki İşletim Sistemi ve Cooja

Yeni nesil ve açık kaynaklı olarak geliştirilen bir işletim sistemi olan Contiki ile IPv6, 6LoWPAN, RPL, CoAP gibi protolkoller Cooja ile simüle edilebilmişlerdir.

Cooja ile istenilen sayıda ve protokolde IoT sensörü eklenebilir, bu sensörlerin kodları değiştirilebilir ve hatta mevcut sensörlerin kodları Cooja'ya eklenebilir. Şekil 3.6'da

Cooja'nın kullanıcı grafik arayüzü görülmektedir. Şekil 3.6'daki simülasyondaki 1 numaralı düğüm, "kök" olarak ayarlanmıştır. Diğer düğümler ise "sensör" dür.




**Şekil 3.1: Cooja'nın Kullanıcı Grafik Arayüzü.**

### 3.1.2 RPL Protokolü Saldırılarına İlişkin Yapılan Saldırıların Simülasyonu İle İlgili Yapılan Çalışmalar

D'Hondt ve diğerlerinin (2015) hazırladıkları akademik rapor ile Cooja IoT simülatörünü kullanarak RPL protokolüne yönelik saldırılardan Taşma Saldırılarını, Sürüm Numarası Artırma Saldırılarını ve Azaltılmış Rank Saldırılarını (Decreased Rank) simüle etmeyi başarmışlardır.

Yapılan bu çalışma ve saldırı simülasyonları ile ilgili düzenlemeler, D'Hondt ve diğerlerinin (2015) oluşturduğu, https://github.com/dhondta/rpl-attacks depoda barınmaktadır.

Tez çalışması için, Github deposunda bulunan talimatlar izlenerek, Linux işletim sistemi olan gerçek bir bilgisayara saldırı senaryolarının bulunduğu işletim sistemi, sanal bir ortama(virtualbox) kurulmuştur.

D'Hondt'un (2015) oluşturduğu simülasyon ile sadece tek zafiyetli düğüm eklenebilmekte, simülasyondaki düğüm sayısı ve zaman parametreleri artırıldığında simülasyonda donmalar meydana gelmektedir. Ayrıca simülasyon güç kullanımı ve düğümlerin birbirleri ile olan ilişkisini içeren görsel veriler oluşturmaktadır. Bu tez

çalışmasında ihtiyacımız olan veri seti olduğu için, Taşma Saldırıları (Flooding Attack), Sürüm Numarası Değiştirme Saldırıları (Version Number Increase Attack) ve Azaltılmış Rank Saldırıları (Decreased Rank) için ihtiyacımız olan zaafiyetli ve normal düğümler (IoT Cihazları) cooja simülatöründe kullanılmak üzere D'Hondt ve diğerleri (2015) tarafından oluşturulan simülasyondan üretilecektir.

### 3.1.3 Saldırı Simülasyon Parametreleri ve Düğümlerin Üretilmesi

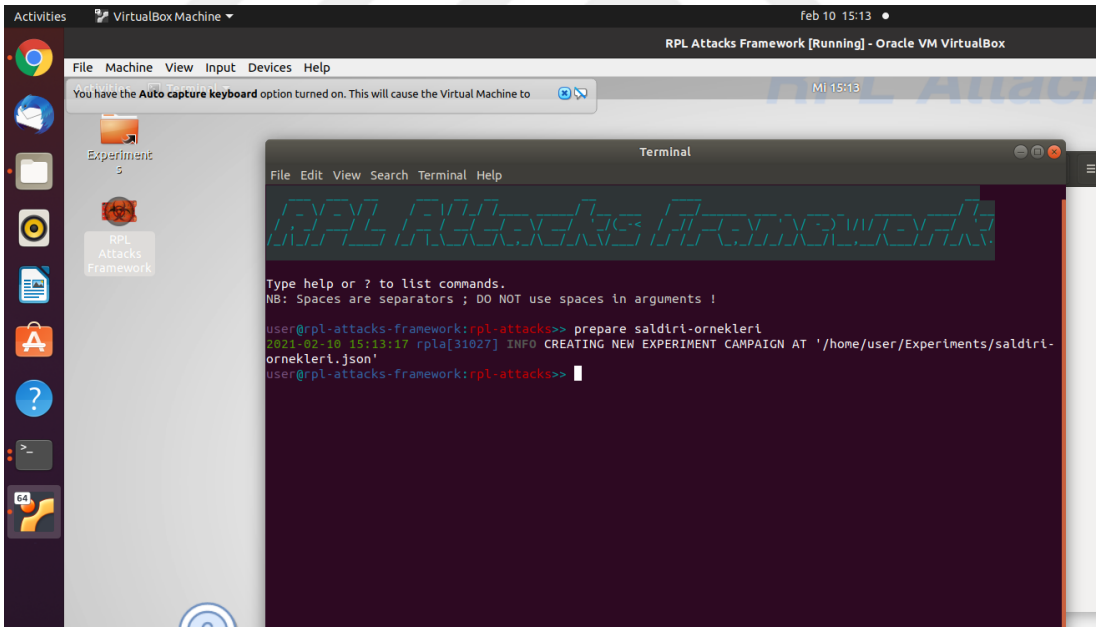
Zafiyetli ve normal düğümleri oluşturmak için aşağıdaki aşamalar izlenmiştir:

#### 3.1.3.1 Saldırı Parametrelerinin Oluşturulması

Sanal makinenin terminaline yazılan

```
user@instant-contiki:rpl-attacks>> prepare saldiri-ornekleri
```

komutu ile saldırı senaryolarının parametrelerinin yer aldığı bir json (JavaScript Object Notation- JavaScript Nesne Notasyonu) dosyası oluşturulmuştur. Sanal makinenin kullanıcı ara yüzü Şekil 3.2'de görülmektedir.




**Şekil 3.2: Parametre Dosyalarının Oluşturulması.**

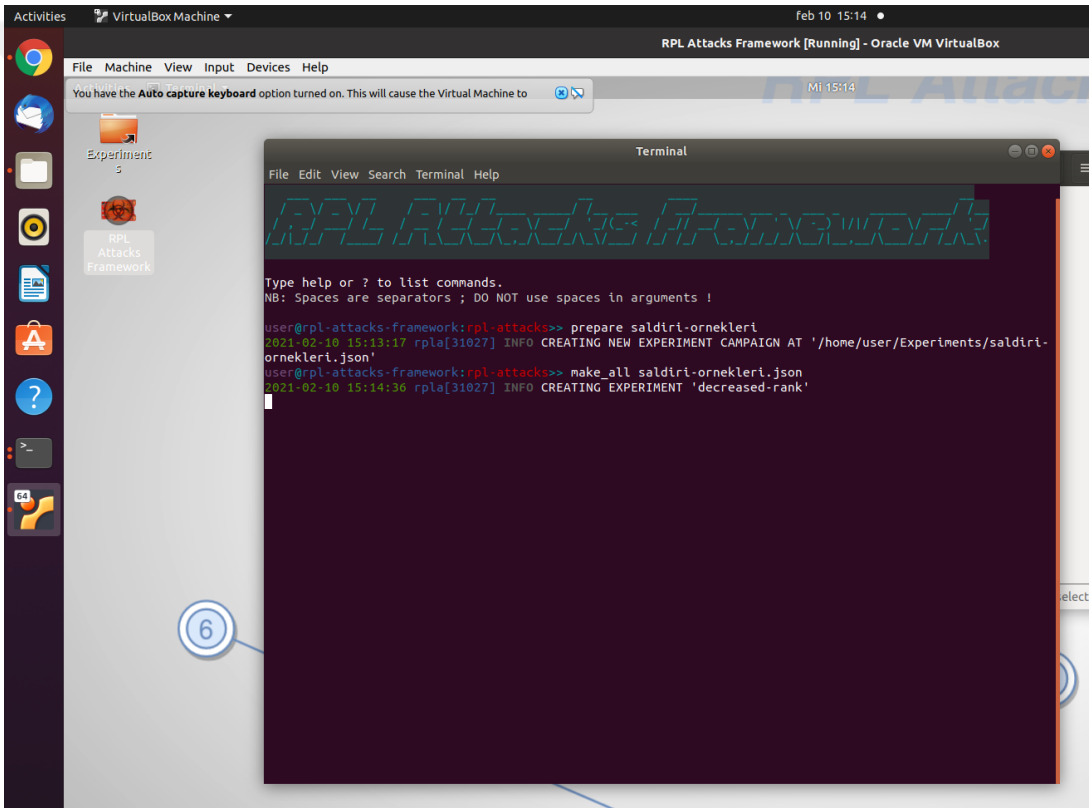
Bu dosya, `~/Experiments` klasörünün içerisine oluşturulur. Bu dosyadaki parametreler değiştirilerek veya yeni parametreler eklenerek simülasyon koşulları belirlenir. Simülatör tarafından oluşturulan `saldiri-ornekleri.json` dosyasının içeriği EK-1'dedir.

### 3.1.3.2 Düğümlerin Oluşturulması

Saldırı parametrelerinin oluşturulması ve içeriğinin değiştirilmesinden sonra,

`user@instant-contiki:rpl-attacks>> make_all saldiri-ornekleri`

komutu terminal ekranına girilmiştir. Simülasyon, `~/Experiments` klasörünün içinde, parametrelerde belirtilen başlıklara göre klasörler oluşturur. Bu klasörlerin her birinin içerisinde, "with-malicious" ve "without-malicious" klasörleri bulunmaktadır. Bu klasörlerin de içindeki "motes" klasöründe 3.2.3.1 maddesinde belirlenen parametrelere göre saldırı klasörleri Şekil 3.3'de oluşturulan saldırı klasörlerinin içeriği Şekil 3.4'dedir. Düğümlerin C kodları incelendiğinde, zafiyetli düğüm oluşturmak için EK-2'deki kod bloklarının eklendiği tespit edilmiştir.

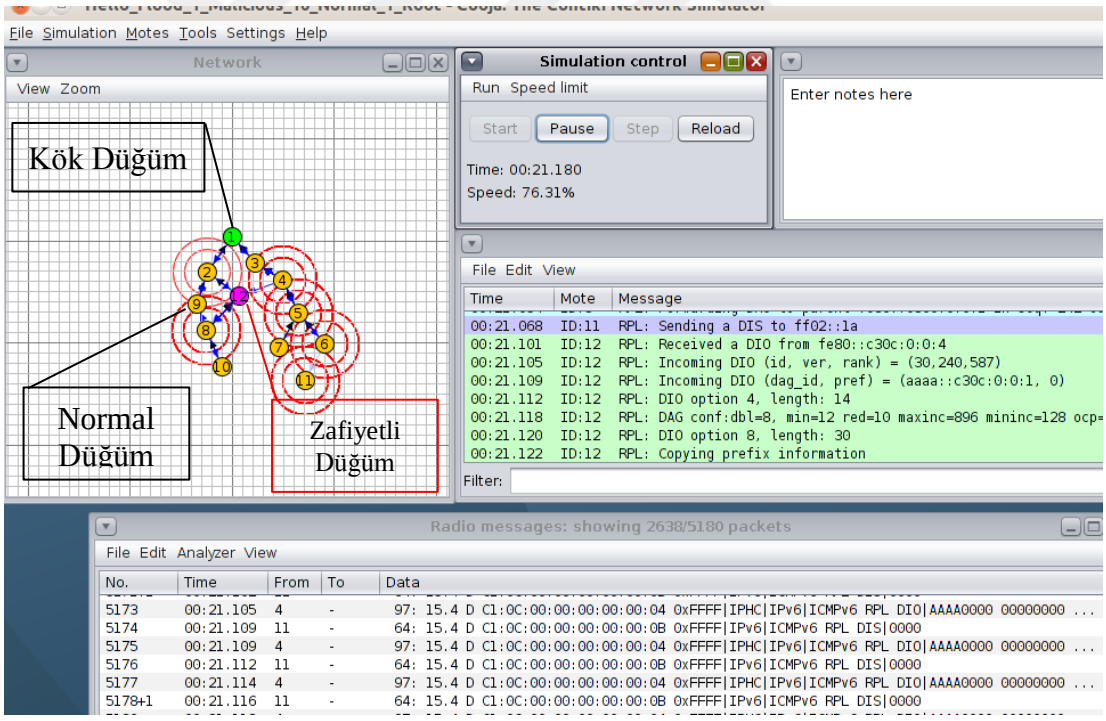



**Şekil 3.3: Make_All Komutu İle Düğümlerin Oluşturulması.**


**Şekil 3.4: Saldırı Türüne Göre Oluşturulan Zaafiyetli ve Normal Düğümler.**

### 3.1.3.3 Oluşturulan Düğümlerle Simülasyonun Yapılması

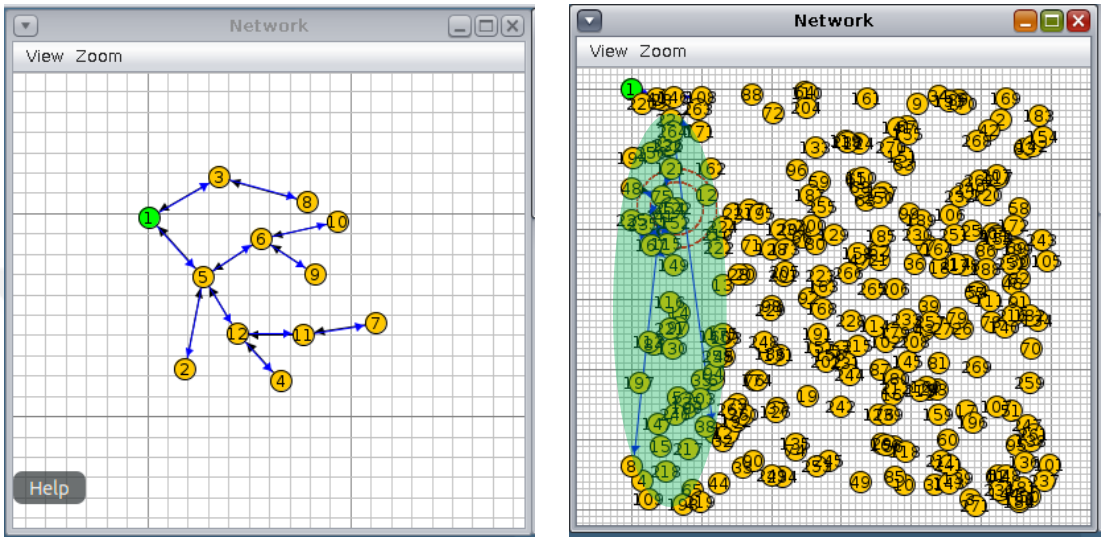
Oluşturulan farklı sayılarda normal ve zafiyetli düğümler, cooja simülatörüne eklenmiştir. Düğümlere ait örnek görsel şekil 3.5'tedir.




**Şekil 3.5: Zafiyetli Düğümle Yapılan Simülasyon.**

Veri setini zenginleştirmek maksadıyla, farklı düğüm sayıları ile simülasyon yapılmıştır. Tablo 3.1'de hangi tür saldırı için kaç adet normal ve zafiyetli düğüm kullanıldığı gösterilmektedir. Ancak, simülasyonda 250 ve daha fazla düğüm kullanıldığında, simülasyonun istikrarlı çalışmadığı tespit edilmiştir. Şekil 3.6'da 10

adet normal düğüm ve 250 adet normal düğüm kullanılarak yapılan simülasyona ait görüntüler yer almaktadır. 10 düğümle yapılan simülasyonda DODAG istenildiği gibi oluşurken, 250 düğümden oluşan simülasyonda düğümler arasındaki iletişim RPL protokolünün doğası gereği "en kısa yol" halinde oluşmadığı görülmüştür. 250 düğümden oluşan senaryoda, düğümlerin çok uzak düğümlere DAO mesajları ilettiği Şekil 3.6'de görülmektedir.




11 düğümden oluşan simülasyonda DODAG düzgün oluşurken, 270 düğümden oluşan senaryoda düğümler çok uzak mesafedeki düğümlerle iletişim kurmaya çalışmaktadır.

**Şekil 3.6: 11 ve 270 Düğümden Oluşan Simülasyon Görüntüleri.**

Ham veri setinin sağlıklı oluşabilmesi için simülasyonlarda toplam 12 düğüm kullanılarak simülasyon 300 saniye çalıştırılmıştır. Simülasyonlara ait zafiyetli, kök ve normal düğüm sayıları tablo 3.1'dedir.

**Tablo 3.1: Simülasyon Düğümlerinin Türleri ve Adetleri.**

| | Normal | | Zafiyetli | | |
|---|---|---|---|---|---|
| **Saldırı Türü** | **Kök** | **Normal** | **Kök** | **Normal** | **Zafiyetli** |
| Taşma Saldırısı | 1 | 10 | 1 | 9 | 1 |
| Azaltılmış Rank | 1 | 10 | 1 | 9 | 1 |
| Sürüm Numarası Artırma Saldırısı | 1 | 10 | 1 | 9 | 1 |

Simülasyon sonunda veriler "pcap" formatında cooja tarafından oluşturulmaktadır. Oluşturulan ".pcap" formatındaki dosyalar, wireshark paket analiz programı ile ".csv" uzantılı dosyalara çevrilmiştir. Oluşturulan ham veri formatı Tablo 3.2'dedir.

**Tablo 3.2: Simülasyon Sonucunda Oluşturulan Ham Veri.**

| No | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0 | fe80::c30c:0:0:c | ff02::1a | ICMPv6 | 64 | RPL Control (DODAG Information Solicitation) |
| 2 | 3270 | fe80::c30c:0:0:c | ff02::1a | ICMPv6 | 64 | RPL Control (DODAG Information Solicitation) |
| 3 | 6565 | fe80::c30c:0:0:c | ff02::1a | ICMPv6 | 64 | RPL Control (DODAG Information Solicitation) |
| ..... | | | | | | |

## 3.2 Ham Verilerin Anlamlı Hale Getirilmesi

Ham veri setinden elde edilen bilgiler, makine öğrenmesini uygulamaya yetmeyecektir. Zafiyetli düğüm içeren simülasyonlardan elde edilen ham veri, tamimiyle normal düğümler içeren simülasyonlardan elde edilen ham veriden farklı olacaktır. Bu farklılığın paket sayısı, mesaj türleri, toplam paket uzunlukları ve oranları olduğu gözlemlenmiştir. Bu anomaliyi tespit etmek için ham veriler 1 saniyelik çerçevelere bölünmüştür. Bu bir saniyelik çerçeveler içinde yeni değerler hesaplanmış yeni aşağıdaki değerler hesaplanarak yeni bir veri seti oluşturulmuştur.

- Kaynak Düğüm: Her düğüm için benzersiz bir sayı oluşturulmuştur.
- Hedef Düğüm: Kaynak düğüm için oluşturulan sayının aynısıdır.
- Paket Sayısı: 1 saniyelik çerçevedeki tüm kaynak düğümlerin sayısıdır.
- Kaynak Düğüm Oranı: (Kaynak Düğüm Sayısı / Paket Sayısı).
- Hedef Düğüm Oranı: (Hedef Düğüm Sayısı / Paket Sayısı).
- Kaynak Düğüm Süresi: 1 saniyelik çerçevede kaynaktan hedefe gönderilen tüm paket sürelerinin toplamıdır.
- Hedef Düğüm Süresi: Hedef tarafından 1 saniyelik çerçevede alınan tüm paket sürelerinin toplamıdır.
- Toplam Paket Süresi: 1 saniyelik çerçevedeki tüm paket sürelerinin toplamıdır.
- Toplam Paket Uzunluğu: 1 saniyelik çerçevedeki tüm paket uzunluklarının toplamıdır.
- Kaynak Paket Oranı: (Kaynak Paket uzunluklarının Toplamı / Toplam Paket Uzunluğu).
- Hedef Paket Oranı: (Hedef Paket uzunluklarının Toplamı / Toplam Paket Uzunluğu).
- DIO Mesaj Sayısı: 1 saniyelik çerçevedeki DIO mesajlarının sayısı.
- DIS Mesaj Sayısı: 1 saniyelik çerçevedeki DIS mesajlarının sayısı.

- DAO Mesaj Sayısı: 1 saniyelik çerçevedeki DAO mesajlarının sayısı
- Diğer Mesaj Sayısı: DIO, DIS ve DAO dışındaki mesajların sayısı.
- Etiket: 0 veya 1 (Zafiyetli veri seti 1, normal veri seti 0'dır).

Yeni veri setinin oluşturulması, Şekil-3.7'deki sözde kod vasıtası ile olmuştur. Yeni oluşan veri setinin örneği Tablo 3.3'dedir. Python kodları EK-3'tedir.

```
START
Dset=INPUT(RawDataset)
WHILE Dset Rows Ends
  Duration=time(current_row)-time(previous_row)
  Duration_list=APPEND(Duration)
ENDWHILE
Dset = Dset + Duration_list
IP_dictionary={IP_Adress :unique_number}
Crr_scnd=60
Counter=0
fs=FLOOR(Dset[Duration_list])
WHILE counter < frame_second
  osf= GET(Dset[Time]>= fs and Dset[Time]<= Crr_scnd+1)
  WHILE osf Rows Ends:
    Osf_list=[ src=IP_dictionary[Source IP_Adress],
               dst=IP_dictionary[Dest. IP_Adress],
               pct_cnt=COUNT(rows)
               src_mote_rt= COUNT(src)/pct_cnt
               dst_mote_rt= COUNT(dst)/pct_cnt
               src_mote_dur=SUM(src_duration)
               dst_mote_dur= SUM(dst_duration)
               ttal_pckt_dur= SUM(duration)
               ttal_pckt_lngth= SUM(pckt_lngth)
               src_pckt_rt= SUM(src_pckt_lngth)/ ttal_pckt_lngth
               dst_pckt_rt= SUM(dst_pckt_lngth)/ ttal_pckt_lngth
               dio_msg_cnt= COUNT(dio_messages)
               dis_msg_cnt= COUNT(dis_messages)
               dao_msg_cnt= COUNT(dao_messages)
               other_msg_cnt= COUNT(other_messages)
               IF Dset="Normal"
                 Label=0
               ELSE
                 Label=1
               ENDIF
  ENDWHILE
New_dset=APPEND(Osf_list)
ENDWHILE
END
```

**Şekil 3.7: Yeni Veri Setinin Oluşturulması İçin Düzenlenen Sözde Kod.**

Yavuz (2018) hazırladığı yüksek lisans tezinde, yukarıdaki özetlemeye benzer bir özetleme yapmıştır. 18 özellik çıkaran Yavuz'un (2018) aksine, bu yüksek lisans tezinde 16 özellik çıkarılmıştır.

**Tablo 3.3: Anlamlı Hale Getirilen Veri Setinin Görünümü.**

| second | src | dst | packetcount | src_ratio | dst_ratio | src_duration_ratio | dst_duration_ratio | TotalPacketDuration | TotalPacketLenght | src_packet_ratio | dst_packet_ratio | DioCount | DisCount | DaoCount | OtherMsg | label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 60 | 6 | 11 | 9 | 0,054545 | 1 | 0,190215 | 1 | 0,22601 | 873 | 0,080409 | 1 | 9 | 0 | 0 | 0 | 1 |
| 60 | 10 | 11 | 156 | 0,945455 | 1 | 0,809785 | 1 | 0,96217 | 9984 | 0,919591 | 1 | 0 | 156 | 0 | 156 | 1 |
| 61 | 4 | 11 | 30 | 0,16129 | 1 | 0,224116 | 1 | 0,23259 | 2910 | 0,225686 | 1 | 30 | 0 | 0 | 0 | 1 |
| 61 | 10 | 11 | 156 | 0,83871 | 1 | 0,775884 | 1 | 0,80521 | 9984 | 0,774314 | 1 | 0 | 156 | 0 | 312 | 1 |
| 62 | 10 | 11 | 156 | 0,83871 | 1 | 0,605037 | 1 | 0,56436 | 9984 | 0,774314 | 1 | 0 | 156 | 0 | 468 | 1 |
| 62 | 5 | 11 | 30 | 0,16129 | 1 | 0,394963 | 1 | 0,36841 | 2910 | 0,225686 | 1 | 30 | 0 | 0 | 0 | 1 |
| …….. | | | | | | | | | | | | | | | | |

Simülasyon esnasında, 12 düğümden oluşan bir sistemin 30'uncu saniyeden sora DODAG yapısının tamamen oluştuğu gözlemlenmiştir. RPL'in doğası gereği DODAG oluşurken cihazlar DIO, DAO ve DAO-ACK mesajlarını birbirlerine gönderecek ve paket trafiği DODAG oluştuktan sonraki trafikten farklı olacaktır. Bu farklılığın makine tarafından öğrenilmesini engellemek için ham veri setinin 60'ıncı saniyesinden sonraki veriler alınarak yeni veri seti oluşturulmuştur.

Her saldırı için oluşturulan ve zafiyetli-normal olarak sınıflandırılan veri setleri, farklı makine öğrenmesi algoritmaları ile karşılaştırılmak için hazır hale gelmiştir. Sonuç olarak Taşma Saldırılarına Ait veri seti, Azaltılmış Rank saldırılarına ait veri seti ve Sürüm Numarası Artırma Saldırılarına ait veri seti olmak üzere toplam 3 adet veri seti oluşturulmuştur.

## 3.3 Makine Öğrenmesi Algoritmalarının Karşılaştırılması

Zafiyetli düğümle yapılan simülasyondan elde edilen veri seti 1 ile, normal düğümlerle yapılan simülasyon 0 ile etiketlendikten sonra bu iki veri seti birleştirilmiştir. Elde edilen bu yeni veri seti ile "sınıflandırma" algoritmaları karşılaştırılacaktır. Karşılaştırılacak makine öğrenmesi algoritmalarının tanımları aşağıda açıklanmıştır.

### 3.3.1 Lojistik Regresyon

Lojistik regresyon için, bir veri setinde sonuca etki eden bir veya birden fazla bağımsız değişken olması gerekmektedir. Lojistik regresyon, bize bir örneklemin bir sınıfa ait olup olmama olasılığını verir. Dolayısıyla lojistik regresyon 1 veya 0 ile etiketlenmiş veriler için uygundur.

Lojistik regresyonun altında yatan matematiksel hesaplama olasılıklara dayanır. Eldeki bağımsız değişkenler, sonucun olup olmamasını etkiliyorsa doğrusal bir denklemle modellenebilir.

Lojistik regresyonu tanımlamak için öncelikle lineer regresyon ile ilgili bilgi vermek gerekir. Y gibi bir bağımlı değişkenin ve x gibi bir bağımsız olduğu bir veri setinde sınıflandırmayı yapabilmek için, denklem 3.1'deki gibi doğrusal bir denklem ile ifade edilebilir.

$$y_i = \beta_0 + \beta_1 x_i + e_i \qquad \textbf{(3.1)}$$

Burada y, bağımlı değişken, $\beta\_0$ kesme terimi ve $\beta\_1$ katsayı(eğim), $e\_i$ hata ve x bağımsız değişken olarak ifade edilebilir. Veri setinde birden fazla bağımsız değişkenin olması durumunda doğrusal fonksiyon denklem 3.2'deki gibi olacaktır:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots . + \beta_n x_n + \varepsilon_i \quad \textbf{(3.2)}$$

Doğrusal denklemlerin sonucunun 0 ile 1 aralığında olmasını istiyorsak, bu denklemleri sigmoid fonksiyonundan geçirmek gerekecektir. Y bağımsız değişkeninin olma olasılığı denklem 3.3'deki gibi olacaktır.

$$P(y) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots . + \beta_n x_n + \varepsilon_i)}} \quad \textbf{(3.3)}$$

Denklem 3.3'deki işleme dayanan lojistik regresyon algoritması, 0 ve 1 ile sınıflandırılmış veri setimiz için uygun bir makine öğrenmesi algoritması olacaktır. (Bonaccorso, 2017)

### 3.3.2 Karar Ağaçları

Karar ağaçları, veri setindeki sınıflandırmayı etkileyen bağımsız değişkenlerin etki katsayısına göre ağaç yapısı ve dallanma oluşturan güçlü bir algoritmadır. Öncelikle veri setinde en güçlü özellik belirlenir, sonrasında ise dallardan birisi seçilir. Bu prosedür, aranılan sınıflandırma hedefine gelene kadar tekrarlanır.

Seçilen bağımsız değişkenler denklem 3.4'deki gibi tanımlanırsa:

$$\sigma = \langle i; t_k \rangle \quad \textbf{(3.4)}$$

Buradaki ilk eleman, veri kümesini belirli bir düğümde bölmek için kullanmak istediğimiz özelliğin dizinidir (yalnızca başlangıçta tüm veri kümesi olacaktır; her adımdan sonra örnek sayısı azalır), ikincisi ise sol ve sağ dalları belirleyen eşiktir.. En iyi eşiğin seçimi, ağacın yapısını ve dolayısıyla performansını belirlediği için temel bir unsurdur. Amaç, örnek veriler ile sınıflandırma sonucu arasında çok kısa bir karar yoluna sahip olmak için en az sayıda bölünmede artık safsızlığı azaltmaktır. Ayrıca iki branşı dikkate alarak bir toplam safsızlık ölçüsü aşağıdaki gibi tanımlanır:

$$I(D, \sigma) = \frac{Nsol}{N_D} I(Dsol) + \frac{Nsağ}{N_D} I(Dsağ) \quad \textbf{(3.5)}$$

Burada, D, seçilen düğümdeki tüm veri kümesidir, Dsol ve Dsağ sonuçtaki alt kümelerdir (seçim demetini uygulayarak) ve I, safsızlık ölçüleridir.

Gini safsızlık endeksi denklem 3.6'daki şekilde tanımlanır:

$$I_{Gini} = \sum p(i|j)\big(1 - p(i|j)\big) \tag{3.6}$$

Burada, toplam her zaman tüm sınıflara genişletilir. Bir veri seti verildiğinde, Gini safsızlığı, dalın olasılık dağılımı kullanılarak rastgele bir etiket seçilirse yanlış sınıflandırma olasılığını ölçer. İndeks, bir düğümün tüm örnekleri tek bir kategoride sınıflandırıldığında minimuma (0.0) ulaşır.

Çapraz entropi ölçüsü denklem 3.7'deki şekliyle tanımlanır:

$$I_{Çapraz-Entropi}(j) = -\sum_{i} p(i|j) \log p(i|j) \tag{3.7}$$

Bu ölçü bilgi teorisine dayanır ve yalnızca tek bir sınıfa ait örnekler bir bölünmede mevcut olduğunda boş değerler alırken, sınıflar arasında tek tip bir dağılım olduğunda maksimumdur (bu, karar ağaçlarındaki en kötü durumlardan biridir çünkü nihai sınıflandırmaya kadar hala birçok karar adımı olduğu anlamına gelir). Bu endeks, Gini safsızlığına çok benzer, ancak daha resmi olarak çapraz entropi, sınıflandırma hakkındaki belirsizliği en aza indiren ayrımı seçmenize izin verirken, Gini safsızlığı yanlış sınıflandırma olasılığını en aza indirir. (Bonaccorso, 2017)

Karar ağaçları, elde edilen veri seti için uygun bir sınıflandırma algoritması olacaktır.

### 3.3.3 Random Forest (Rastgele Orman)

Rastgele orman, bir düğümü bölmek için farklı bir politikaya sahip rastgele örnekler üzerine inşa edilen bir dizi karar ağacıdır: Böyle bir modelde en iyi seçeneği aramak yerine, rastgele bir özellik alt kümesi (her ağaç için) kullanılır. Verileri en iyi ayıran eşiği bulur. Sonuç olarak, daha zayıf bir şekilde eğitilmiş birçok ağaç olacak ve her biri farklı bir tahmin üretecektir. Bu sonuçları yorumlamanın iki yolu vardır; daha yaygın yaklaşım çoğunluk oylamasına dayanır (en çok oylanan sınıf doğru kabul edilecektir). Teorik olarak farklı olsalar bile, eğitilmiş rastgele bir ormanın olasılık ortalaması, tahminlerin çoğundan çok farklı olamaz (aksi takdirde, farklı sabit noktalar olmalıdır); bu nedenle iki yöntem genellikle karşılaştırılabilir sonuçlara götürür.

Özellik önemi kavramı, ormandaki tüm ağaçların ortalamasını hesaplayarak rastgele ormanlara da denklem 3.8'deki gibi uygulanabilir: (Bonaccorso, 2017)

$$\text{Özellik Önemi}(x_i) = \frac{1}{N_{\text{Ağaçlar}}} \sum \sum \frac{N_k}{N} \Delta I_{x_i} \quad \textbf{(3.8)}$$

Rastgele Orman (Random Forest) da veri setimizi eğitmek için uygun bir makine öğrenmesi algoritmasıdır.

### 3.3.4 Navie Bayes

Naive Bayes, Bayes teoremini kullanarak bir dizi koşul verilen bir sonucun olasılığını belirleyen güçlü ve eğitilmesi kolay bir sınıflandırıcı ailesidir. Bayes teoremine göre, koşullu olasılıklar tersine çevrilir, böylece sorgu ölçülebilir büyüklüklerin bir fonksiyonu olarak ifade edilebilir. Bayes teoremine göre iki olasılıklı olay A ve B'de, marjinal olasılıklar P (A) ve P (B) ile koşullu olasılıklar P (A | B) ve P (B | A) ile çarpım kuralını kullanarak denkelm 3.9'daki gibi bir korelasyon kurulabilir.

$$\begin{cases} P(A \cap B) = P(A|B)P(B) \\ P(B \cap A) = P(B|A)P(A) \end{cases} \quad \textbf{(3.9)}$$

Kesişimin değişmeli olduğu düşünülürse, ilk üyeler eşittir; böylece Bayes teoremi denklem 3.10'da olduğu gibi türetilebilir:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \textbf{(3.10)}$$

Bu formülün çok derin felsefi sonuçları vardır ve istatistiksel öğrenmenin temel bir unsurudur. (Bonaccorso, 2017)

Navie bayes algoritması da veri setimiz için uygun bir makine öğrenmesi algoritmasıdır.

### 3.3.5 KNN-K En Yakın Komşu

KNN algoritması, benzer sınıfların birbirine yakın olduğunu varsayar. Bu algoritmanın olumlu yanları, uygulaması kolay, gürültülü verilere karşı dirençli ve büyük eğitim setlerinde daha etkin sonuçlar üretmesi olarak sıralanabilir. Ancak, algoritma başlangıcında "k" parametresine ihtiyaç duyması, en iyi sonucun elde edilebilmesi için hangi uzaklık ölçü biriminin seçileceğinin belirgin olmaması ve hesaplama maliyetinin

yüksekliği olumsuz yanlarıdır. (Bilgin & Yılmaz, 2018) Başka bir deyişle, benzer şeyler birbirine yakındır. Her noktanın en yakın komşuları basit çoğunluk oyuyla hesaplanır. Genellikle bu oy çoğunluğu Öklid mesafesidir.(Denklem 3.11)

$$|AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \textbf{(3.11)}$$

Bir sorgu noktası, noktanın en yakın komşuları içinde en çok temsilciye sahip olan veri sınıfına atanır. Her sorgu noktasının en yakın komşuları, burada kullanıcı tarafından belirtilen bir ***k*** tamsayı değeridir. Burada ***k*** değerinin optimum seçimi önem arz eder ve veriye bağlıdır. ***k*** tamsayı değeri büyüdükçe gürültü artar, ancak sınıflandırma daha belirginleşir. (Bonaccorso, 2017)

KNN sınıflandırma algoritması veri seti için uygun bir sınıflandırma algoritmasıdır.

### 3.3.6 Yapay Sinir Ağları

Yapay bir sinir ağı (YSA), bir giriş katmanını bir çıktıya bağlayan yönlendirilmiş bir yapıdır. Yapay sinir ağlarının genel vektör fonksiyonu denklem 3.12'deki gibidir:

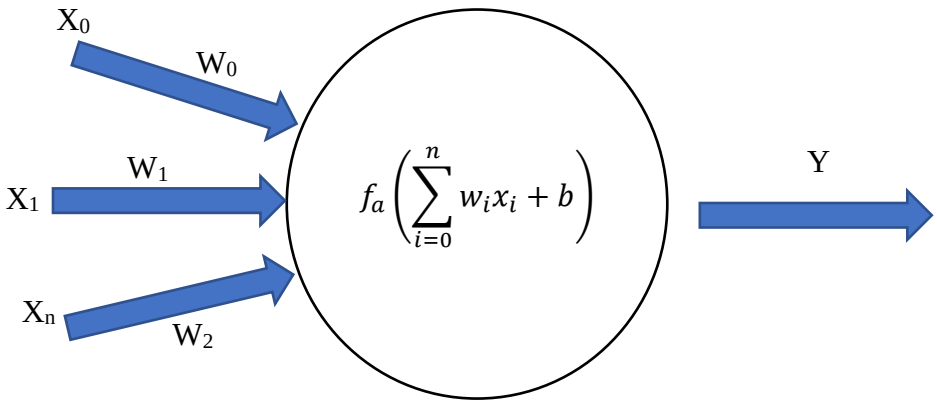
$$\bar{y} = f(\bar{x}) \quad \textbf{(3.12)}$$

Burada,

$$\bar{x} = (x_1 + x_2 + x_3 +, \dots, +x_n) \; ve \; \bar{y} = (y_1 + y_2 + y_3 +, \dots, +y_m) \quad \textbf{(3.13)}$$

"Nöral" kavramı iki önemli unsurdan meydana gelir: temel hesaplama biriminin iç yapısı ve bunlar arasındaki bağlantılardır. Genel anlamda yapay sinir ağlarının şematik gösterimi Şekil 3.8'de verilmiştir.

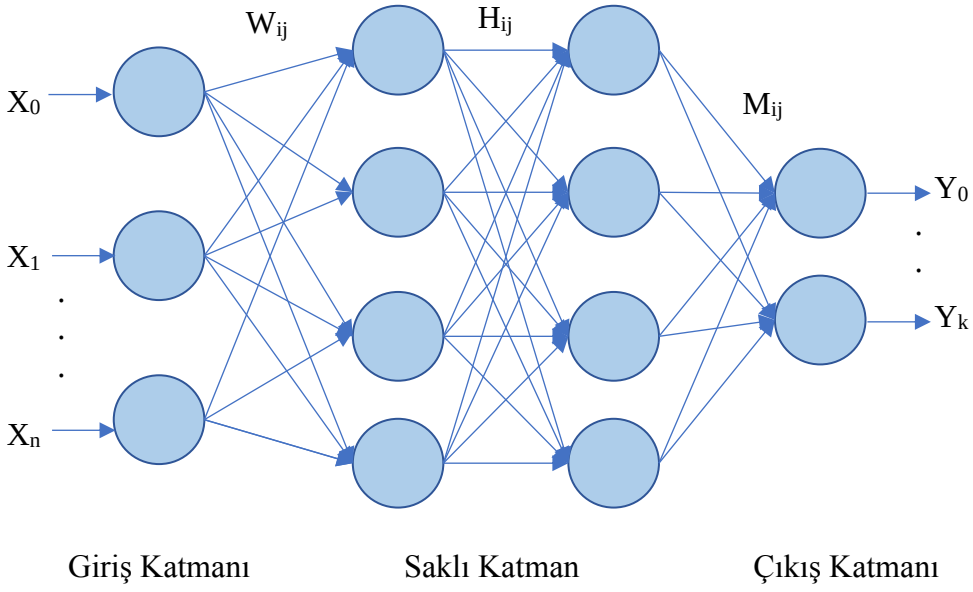
Bir nöron çekirdeği, her biri sinaptik bir ağırlık $w_i$ ile karakterize edilen n giriş kanalına bağlanır. Girdi, bileşenlerine bölünür ve karşılık gelen ağırlık ile çarpılır ve toplanır. Bu toplama isteğe bağlı bir bayes eklenebilir (üniter bir girişe bağlı başka bir ağırlık gibi çalışır). Ortaya çıkan toplam, $f_a$ aktivasyon fonksiyonu tarafından filtrelenir ve çıktı üretilir. Yapay sinir ağları, lojistik regresyon gibi doğrusal modelden farklı ve daha etkili bir yol izlemek için ilk Çok Katmanlı Algılayıcıyı oluşturmak gerekir.




**Şekil 3.8: Yapay Sinir Ağlarının Şematik Gösterimi.**

G. Bonaccorso, Machine Learning Algorithms: A Reference Guide to Popular Algorithms for Data Science and Machine Learning, Packt Publishing, Birmingham, UK, 2017

Aşağıdaki şekilde, bir Çok Katmanlı Algılayıcıyı'nın n boyutlu bir girdiye, p gizli nöronlara ve bir k boyutlu çıktıya sahip şematik bir temsili şekil de gösterilmiştir:




**Şekil 3.9: Çok Boyutlu Yapay Sinir Ağlarının Gösterimi.**

Şekil 3.9'da 4 katman gösterilmiştir. $X_1$ .. $X_n$ giriş vektörlerini alan giriş katmanı; gizli iki katman; ve çıktıyı üretmekten sorumlu olan çıktı $Y_0$….$Y_k$ çıktı katmanı. Şekilde görüldüğü gibi, her nöron bir sonraki katmana ait olan tüm nöronlara bağlıdır ve 3 adet ağırlık matrisi bulunmaktadır. Görülgüğü gibi, her nöron bir sonraki katmana ait tüm nöronlara bağlıdır ve ilk indeksin önceki katmana ve ikincisinin de sonraki katmana

atıfta bulunulduğu kuralı kullanarak, $W_{ij}$, $H_{ij}$ ve $M_{ij}$ olmak üzere üç ağırlık matrisi bulunmaktadır.

Bu nedenle, her gizli nöronun net girdisi ve karşılık gelen çıktı değeri denklem 3.14'deki gibi hesaplanır.

$$\begin{cases} z_j^{Giriş} = W_{0j}X_0 + W_{1j}X_1 + \cdots + W_{nj}X_n = \sum_i W_{ij}X_i \\ z_j^{Çıkış} = f_a^{Saklı\ Kayman}\left(z_j^{Giriş} + b_j^{Saklı\ Katman}\right) \end{cases} \quad \textbf{(3.14)}$$

Aynı şekilde ağ çıktısı denklem 3.15'deki gibi hesaplanır:

$$\begin{cases} y_k^{Giriş} = h_{0k}z_0^{Çıkış} + h_{1k}z_1^{Çıkış} + \cdots + h_{pk}z_p^{Çıkış} = \sum_i h_{jk}z_j^{Çıkış} \\ y_k^{Çıkış} = f_a^{SÇıkış}\left(y_k^{Giriş} + b_k^{Çıkış}\right) \end{cases} \quad \textbf{(3.15)}$$

Görüldüğü gibi, ağ son derece doğrusal olmayan hale getirilmiş ve bu özellik, doğrusal yöntemlerle yönetilmesi imkânsız olan karmaşık senaryoların modellenmesine izin vermiştir. Ancak tüm sinaptik ağırlıklar ve bayes değerlerinin belirlenmesi $f_a$ (x) 'in farklılaştırılabilir olması ile gerçekleştirilir.

Öncelikle bir hata (kayıp) fonksiyonu toplam hata karesi olarak tanımlanır ve bu işlem tüm katmanlara uygulanır.(Denklem 3.16)

$$L = \frac{1}{2}\sum_n \left\|\bar{y}_n^{Tahmin\ Edilen} - \bar{y}_n^{Hedef}\right\|^2 \quad \textbf{(3.16)}$$

Bu fonksiyon tüm değişkenlere bağlıdır (ağırlıklar ve bayes değerleri), akabinde geriye doğru hesaplama yapılır. Bu işlem değişimin ne kadar olduğunun hesaplanması yani türev fonksiyonudur. Tek bir katman için bu fonksiyon denklem deki gibidir. Bu denklem tüm katmanlara genişletilir.(Denklem 3.17)

$$\frac{\partial L}{\partial h_{jk}} = \sum_n \frac{\partial f_a^{Çıkış}}{\partial y_k^{Giriş}}\frac{\partial y_k^{Giriş}}{\partial h_{jk}} = \sum_n \partial_k z_j^{Çıkış}\frac{\partial f_a^{Çıkış}}{\partial y_k^{Giriş}} = \sum_n a_k\, z_j^{Çıkış} \quad \textbf{(3.17)}$$

Burada, alfa terimi (hata deltasıyla orantılıdır) çıktı katmanından gizli olana geri yayılır. Çok sayıda gizli katman varsa, bu prosedür ilk katmana kadar yinelemeli olarak tekrarlanır. Algoritma, gradyan iniş yöntemini benimser; bu nedenle

yakınsamaya kadar ağırlıkları yinelemeli olarak günceller. Böylelikle eldeki veri ile makine öğrenimi gerçekleştirilir. (Bonaccorso, 2017)

Yapay sinir ağları, çok karmaşık sınıflandırmaları yapmamızı sağlar. Çok güçlü bir algoritmadır. Yalnız yapay sinir ağındaki eğitim süresi diğer algoritmalara göre daha uzundur. Daha fazla işlem gerektirir. Bu tez çalışmasında elde edilen veri setini öğrenmesi ve saldırıyı tespit etmesi çok muhtemeldir. Fakat bu tez çalışmasındaki maksat, basit ve hızlı bir yöntem belirlemektir.

## 3.4 Deneylerin Yapılması ve Deney Sonuçları

### 3.4.1 Anlamlı Hale Getirilen Veri Setinin Makine Öğrenmesi ile Analizi

Ham veri setleri anlamlı hale getirilip, normal ve zafiyetli olarak sınıflandırılmıştır. Makine öğrenmesi gerçekleştirilmeden önce de normal ve zafiyetli veri setlerinden eşit miktarda satır alınmıştır. Bunun için de, 60'ıncı saniyeden itibaren elde edilen normal veya zafiyetli veri setlerinden en az satıra sahip olan veri setinin satır sayısı baz alınmış, diğer veri setinden de aynı miktarda satış sayısı alınmıştır. Böylelikle dengeli veri miktarının elde edilmesi hedeflenmiştir. Makinenin kaynak ve hedef IP adreslerine göre bir saldırı olup olmadığını öğrenmemesi için normalizasyon işleminden önce veri setlerinde kaynak ve hedef IP adresleri çıkarılmıştır. Akabinde veriler normalize edilmiştir. Test ve eğitim veri setleri normalizasyon işlemi denklem 3.18 ile yapılmıştır. Burada z, normalleştirilmiş değeri temsil eder, x normalize edilmemiş veridir, μ tüm x değerlerinin ortalamasıdır ve σ, x değerlerinin standart sapmasıdır. Bu işlem, pythondaki StandardScaler kütüphanesi ile gerçekleştirilmiştir.

$$z = \frac{x - \mu}{\sigma} \quad and\ \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)\ \ \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x - \mu)^2} \tag{3.18}$$

Akabinde, veri kümesi 2/3 miktarında test ve eğitim veri kümelerine bölünmüştür. (2/3 eğitim, 1/3 test).

Bu aşamadan sonra, veri setleri farklı makine öğrenmesi algoritmaları ile eğitilip, test edilerek RPL protokolünde gerçekleştirilecek Taşma, Sürüm Numarası Artırma ve Azaltılmış Rank saldırılarının tespitinde kullanılacak en uygun makine öğrenmesi algoritması belirlenecektir. Bu maksatla, altı tür makine öğrenimi algoritması test

edilmiştir. Bunlar Lojistik Regresyon Sınıflandırması, Karar Ağaçları, Rastgele Orman, Navie Bayes, KNN Sınıflandırıcı ve Yapay Sinir Ağları algoritmalarıdır. Deneyler, 16 GB RAM, Intel(R) Core(TM) i7-8565U CPU 1.80GHz 1.99 GHz , özelliklerine sahip bilgisayar ile gerçekleştirilmiştir. Makine öğrenmesi algoritmalarının uygulaması Python 3 ile icra edilmiştir. Makine öğrenimi için uygulanan parametreler Tablo-3.4'de gösterilmektedir.

**Tablo 3.4: Makine Öğrenmesi Algoritmalarında Kullanılan Parametreler.**

| Algoritma | Parametre |
|---|---|
| **Lojistik Regresyon** | **Kütüphane:** sklearn.linear_model, LogisticRegression<br>Kütüphanenin mevcut parametreleri kullanılmıştır. |
| **Rastgele Orman** | **Kütüphane:** sklearn.ensemble, RandomForestClassifier<br>**Parametreler:** n_estimators=8, criterion='entropy' |
| **Karar Ağaçları** | **Kütüphane:** sklearn.tree, DecisionTreeClassifier<br>**Parametreler:** criterion='entropy' |
| **Navie Bayes** | **Kütüphane:** sklearn.naive_bayes, GaussianNB<br>Kütüphanenin mevcut parametreleri kullanılmıştır. |
| **KNN Sınıflandırıcı** | **Kütüphane:** sklearn.neighbors, KNeighborsClassifier<br>Kütüphanenin mevcut parametreleri kullanılmıştır. |
| **Yapay Sinir Ağları** | **Kütüphane:** tensorflow, keras<br>6 Katmanlı bir yapı uygulanmıştır.<br>Neron sayıları:26,52,56,13,7,1<br>Optimizer:”Nadam”; Tekrar:60; Tahmin Eşiği:0,7 |

Deneylerin icra edilmesinden sonra doğruluk oranı ve eğitim süresi değerleri üzerinden karşılaştırılacaktır.

Doğruluk oranı (DO), denklem 3.19’daki gibi hesaplanmış ve GP Gerçek Pozitif, GN Gerçek Negatif, YP Yanlış Pozitif ve YN yanlış negatiftir.

$$DO = \frac{GP + GN}{GP + GN + YP + YN} \quad \textbf{(3.19)}$$

Deneyler icra edildikten sonra, Taşma saldırısı, Sürüm Numarası Artırma Saldırısı, Azaltılmış Rank Saldırısına ait veri setlerinin kaç satırdan oluştuğu, doğruluk oranı ve eğitim süresi değerleri Tablo 3.5’te gösterilmiştir. Veri setinin makine öğrenmesi algoritmaları ile eğitilmesinden sonra, elbette hata oranının tespiti daha kısa zaman alacaktır. Ancak, bu tez çalışmasında, aynı saldırılara ait veri setleri farklı makine

öğrenmesi algoritmaları ile analiz edilip bu algoritmalar arasındaki en hızlı, etkili, karmaşık olmayan ve güvenilir algoritmayı tespit etmek amaçlanmıştır. Dolayısıyla daha kısa sürede eğitimi gerçekleştiren ve yüksek doğrulukla saldırıyı tespit edebilen algoritma daha maliyet etkin olacaktır. Aşağıda saldırılara ait elde edilen sonuçlar yorumlanmıştır.

**Tablo 3.5: Deney Sonuçları.**

| Saldırı Türü | Veri Seti Satır Sayısı | Test ve Eğitim Veri Seti Satır Sayısı | Algoritma | Doğruluk Oranı(%) | Eğitim Süresi (ms) |
|---|---|---|---|---|---|
| Taşma Saldırısı (Flooding Attack) | N:214 Z:1259 T:1473 | N:214 Z:214 T:428 | Lojistik Regresyon | 95,7 | 363 |
| | | | Karar Ağaçları | 93,6 | 0 |
| | | | Rastgele Orman | 95,0 | 168 |
| | | | Navie Bayes | 69,7 | 13 |
| | | | K En Yakın Komşu | 95,7 | 4 |
| | | | Yapay Sinir Ağları | 97,2 | 1847 |
| Sürüm Numarası Artırma Saldırısı | N:168 Z:1114 T:1282 | N:168 Z:168 T:336 | Lojistik Regresyon | 74,8 | 185 |
| | | | Karar Ağaçları | 74,8 | 2 |
| | | | Rastgele Orman | 72,9 | 253 |
| | | | Navie Bayes | 74,8 | 5 |
| | | | K En Yakın Komşu | 81,0 | 2 |
| | | | Yapay Sinir Ağları | 72,0 | 1688 |
| Azaltılmış Rank Saldırısı | N:160 Z:151 T:311 | N:151 Z:151 T:302 | Lojistik Regresyon | 54,0 | 5 |
| | | | Karar Ağaçları | 57,0 | 2 |
| | | | Rastgele Orman | 58,0 | 10 |
| | | | Navie Bayes | 54,0 | 1 |
| | | | K En Yakın Komşu | 56,0 | 0 |
| | | | Yapay Sinir Ağları | 58,0 | 1720 |

Tablo Açıklamaları: N : Normal, Z : Zafiyetli, T : Toplam

### 3.4.2 Anlamlı Hale Getirilen Veri Setinin İstatistiksel Analizi

Tablo 3.6'da Normal Düğümle Yapılan Azaltılmış Rank Deneyi, Normal Düğümle yapılan Taşma Deneyi, Normal Düğümle Yapılan Sürüm Numarası Artırma Deneyi, Zafiyetli Düğümle Yapılan Azaltılmış Rank Deneyi, Zafiyetli Düğümle Yapılan Taşma Saldırısı Deneyi ve Zafiyetli Düğümle Yapılan Sürüm Numarası Artırma Deneyine ait anlamlandırılmış veri tablolarına ait ortalama, medyan, maksimum, minimum, standart sapma, çarpıklık ve basıklık değerleri gözükmektedir. Elde edilen bu verilerle, anomalinin yorumlanması amaçlanmıştır.

**Tablo 3.6: Anlamlı Hale Gelen Verilerin İstatistiksel Değerleri.**

| | | Paket Sayısı | Kaynak Düğüm Oranı | Hedef Düğüm Oranı | Kaynak Düğüm Süresi | Hedef Düğüm Süresi | Toplam Paket Süresi | Toplam Paket Uzunluğu | Kaynak Paket Oranı | Hedef Paket Oranı | DIO Sayısı | DIS Sayısı | DAO Sayısı | Diğer Mesaj Sayısı |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AzRnkNrm | Ortalama | 25,981 | 0,561 | 0,578 | 0,563 | 0,581 | 0,719 | 2244,019 | 0,562 | 0,579 | 11,913 | 0,000 | 14,069 | 0 |
| TaSalNrm | Ortalama | 21,299 | 0,626 | 0,643 | 0,627 | 0,643 | 0,900 | 1900,252 | 0,627 | 0,643 | 12,364 | 0,000 | 8,935 | 0 |
| SNANrm | Ortalama | 21,315 | 0,673 | 0,688 | 0,673 | 0,690 | 1,017 | 1886,964 | 0,673 | 0,689 | 11,780 | 0,000 | 9,536 | 0 |
| AzRnkZf | Ortalama | 25,921 | 0,624 | 0,650 | 0,626 | 0,649 | 0,918 | 2307,656 | 0,624 | 0,650 | 14,517 | 0,000 | 11,404 | 0 |
| TaSalZf | Ortalama | 81,478 | 0,433 | 0,910 | 0,433 | 0,912 | 0,417 | 5713,654 | 0,433 | 0,907 | 14,168 | 65,523 | 1,786 | 17375,458 |
| SNAZf | Ortalama | 34,320 | 0,440 | 0,608 | 0,440 | 0,623 | 0,335 | 2975,820 | 0,440 | 0,613 | 17,218 | 0,000 | 17,101 | 0 |
| AzRnkNrm | Medyan | 24,000 | 0,500 | 0,500 | 0,529 | 0,523 | 0,089 | 2217,500 | 0,500 | 0,502 | 0,000 | 0,000 | 3,000 | 0 |
| TaSalNrm | Medyan | 23,000 | 0,577 | 0,634 | 0,668 | 0,727 | 0,079 | 1824,000 | 0,598 | 0,629 | 5,000 | 0,000 | 0,000 | 0 |
| SNANrm | Medyan | 22,000 | 0,806 | 0,938 | 0,959 | 0,970 | 0,086 | 2040,000 | 0,806 | 0,934 | 7,000 | 0,000 | 0,000 | 0 |
| AzRnkZf | Medyan | 21,000 | 0,576 | 0,663 | 0,689 | 0,754 | 0,080 | 1746,000 | 0,579 | 0,622 | 5,000 | 0,000 | 0,000 | 0 |
| TaSalZf | Medyan | 30,000 | 0,161 | 1,000 | 0,333 | 1,000 | 0,283 | 2910,000 | 0,226 | 1,000 | 6,000 | 0,000 | 0,000 | 0 |
| SNAZf | Medyan | 30,000 | 0,378 | 0,563 | 0,341 | 0,689 | 0,129 | 2910,000 | 0,386 | 0,599 | 26,000 | 0,000 | 0,000 | 0 |
| AzRnkNrm | Maksimum | 259,000 | 1 | 1 | 1 | 1 | 10,311 | 19684,000 | 1 | 1 | 63,000 | 0,000 | 259,000 | 0 |
| TaSalNrm | Maksimum | 94,000 | 1 | 1 | 1 | 1 | 16,991 | 7144,000 | 1 | 1 | 49,000 | 0,000 | 94,000 | 0 |
| SNANrm | Maksimum | 96,000 | 1 | 1 | 1 | 1 | 16,543 | 7296,000 | 1 | 1 | 30,000 | 0,000 | 96,000 | 0 |
| AzRnkZf | Maksimum | 214,000 | 1 | 1 | 1 | 1 | 24,457 | 21828,000 | 1 | 1 | 214,000 | 0,000 | 160,000 | 0 |
| TaSalZf | Maksimum | 278,000 | 1 | 1 | 1 | 1 | 1,221 | 20094,000 | 1 | 1 | 197,000 | 278,000 | 246,000 | 82494 |
| SNAZf | Maksimum | 286,000 | 1 | 1 | 1 | 1 | 3,386 | 21736,000 | 1 | 1 | 200,000 | 0,000 | 286,000 | 0 |

**Tablo 3.6-devam.**

| | | Paket Sayısı | Kaynak Düğüm Oranı | Hedef Düğüm Oranı | Kaynak Düğüm Süresi | Hedef Düğüm Süresi | Toplam Paket Süresi | Toplam Paket Uzunluğu | Kaynak Paket Oranı | Hedef Paket Oranı | DIO Sayısı | DIS Sayısı | DAO Sayısı | Diğer Mesaj Sayısı |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AzRnkNrm | Minimum | 1 | 0,004 | 0,041 | 0,002 | 0,011 | 0,002 | 102,000 | 0,004 | 0,036 | 0 | 0 | 0 | 0 |
| TaSalNrm | Minimum | 1 | 0,015 | 0,015 | 0,007 | 0,007 | 0,003 | 76,000 | 0,013 | 0,013 | 0 | 0 | 0 | 0 |
| SNANrm | Minimum | 2 | 0,031 | 0,031 | 0,010 | 0,015 | 0,006 | 152,000 | 0,030 | 0,030 | 0 | 0 | 0 | 0 |
| AzRnkZf | Minimum | 2 | 0,037 | 0,037 | 0,001 | 0,001 | 0,006 | 152,000 | 0,034 | 0,034 | 0 | 0 | 0 | 0 |
| TaSalZf | Minimum | 1 | 0,004 | 0,007 | 0,000 | 0,008 | 0,000 | 97,000 | 0,005 | 0,007 | 0 | 0 | 0 | 0 |
| SNAZf | Minimum | 1 | 0,010 | 0,015 | 0,000 | 0,003 | 0,002 | 76,000 | 0,011 | 0,015 | 0 | 0 | 0 | 0 |
| AzRnkNrm | Std.S. | 31,682 | 0,336 | 0,345 | 0,356 | 0,357 | 1,822 | 2458,645 | 0,338 | 0,345 | 14,465 | 0,000 | 33,643 | 0 |
| TaSalNrm | Std.S. | 11,577 | 0,354 | 0,359 | 0,369 | 0,371 | 2,321 | 1051,003 | 0,355 | 0,360 | 13,544 | 0,000 | 13,137 | 0 |
| SNANrm | Std.S. | 12,321 | 0,353 | 0,348 | 0,368 | 0,360 | 2,734 | 1047,783 | 0,352 | 0,348 | 12,659 | 0,000 | 14,750 | 0 |
| AzRnkZf | Std.S. | 32,385 | 0,347 | 0,344 | 0,378 | 0,373 | 2,565 | 2921,495 | 0,349 | 0,346 | 26,173 | 0,000 | 26,402 | 0 |
| TaSalZf | Std.S. | 66,875 | 0,367 | 0,236 | 0,338 | 0,238 | 0,339 | 4041,383 | 0,334 | 0,234 | 18,604 | 77,826 | 14,573 | 25605,433 |
| SNAZf | Std.S. | 40,048 | 0,279 | 0,330 | 0,334 | 0,363 | 0,497 | 3042,565 | 0,279 | 0,333 | 15,638 | 0,000 | 44,144 | 0 |
| AzRnkNrm | Çarpıklık | 4,644 | 0,102 | 0,096 | 0,028 | 0,005 | 3,622 | 4,084 | 0,089 | 0,081 | 0,665 | - | 4,787 | - |
| TaSalNrm | Çarpıklık | 1,003 | -0,163 | -0,256 | -0,254 | -0,319 | 3,878 | 0,449 | -0,177 | -0,271 | 0,411 | - | 2,118 | - |
| SNANrm | Çarpıklık | 1,505 | -0,381 | -0,488 | -0,463 | -0,566 | 3,680 | 0,768 | -0,375 | -0,482 | 0,419 | - | 2,204 | - |
| AzRnkZf | Çarpıklık | 4,118 | -0,157 | -0,273 | -0,342 | -0,444 | 6,036 | 4,580 | -0,180 | -0,296 | 5,499 | - | 4,492 | - |
| TaSalZf | Çarpıklık | 0,312 | 0,388 | -2,735 | 0,432 | -2,829 | 0,583 | 0,323 | 0,478 | -2,630 | 3,118 | 0,385 | 11,488 | 1,187 |
| SNAZf | Çarpıklık | 4,589 | 0,769 | -0,070 | 0,491 | -0,337 | 2,684 | 4,376 | 0,777 | -0,126 | 1,521 | - | 4,324 | - |

**Tablo 3.6-devam.**

| | | Paket Sayısı | Kaynak Düğüm Oranı | Hedef Düğüm Oranı | Kaynak Düğüm Süresi | Hedef Düğüm Süresi | Toplam Paket Süresi | Toplam Paket Uzunluğu | Kaynak Paket Oranı | Hedef Paket Oranı | DIO Sayısı | DIS Sayısı | DAO Sayısı | Diğer Mesaj Sayısı |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AzRnkNrm | Basıklık | 26,453 | -1,405 | -1,528 | -1,558 | -1,582 | 13,663 | 22,247 | -1,427 | -1,519 | -0,836 | - | 26,461 | - |
| TaSalNrm | Basıklık | 6,635 | -1,553 | -1,544 | -1,581 | -1,573 | 17,348 | 1,858 | -1,550 | -1,539 | -1,548 | - | 8,087 | - |
| SNANrm | Basıklık | 7,563 | -1,539 | -1,413 | -1,510 | -1,360 | 13,942 | 2,994 | -1,547 | -1,424 | -1,568 | - | 7,281 | - |
| AzRnkZf | Basıklık | 18,334 | -1,520 | -1,495 | -1,501 | -1,405 | 48,539 | 24,913 | -1,506 | -1,475 | 39,111 | - | 22,105 | - |
| TaSalZf | Basıklık | -1,670 | -1,602 | 6,238 | -1,302 | 6,673 | -1,018 | -1,328 | -1,335 | 5,722 | 20,477 | -1,754 | 149,272 | -0,070 |
| SNAZf | Basıklık | 22,685 | -0,261 | -1,435 | -1,174 | -1,479 | 8,003 | 21,355 | -0,252 | -1,432 | 16,123 | - | 20,259 | - |

**Tablo Açıklamaları:**

AzRnkNrm : Normal Düğümle Yapılan Azaltılmış Rank Deneyi,
TaSalNrm : Normal Düğümle yapılan Taşma Deneyi,
SNANrm : Normal Düğümle Yapılan Sürüm Numarası Artırma Deneyi,
AzRnkZf : Zafiyetli Düğümle Yapılan Azaltılmış Rank Deneyi,
TasalZf : Zafiyetli Düğümle Yapılan Taşma Saldırısı Deneyi,
SNAZf : Zafiyetli Düğümle Yapılan Sürüm Numarası Artırma Deneyi,
Std.S. : Standart Sapma.

#### 3.4.2.1 Ortalama Değerlerinin İncelenmesi

Taşma saldırılarında ve sürüm numarası artırma saldırılarında, paket sayıları ortalamalarında bariz bir artış gözlemlenmiştir. Normal düğümlere ait paket ortalamaları 21-25 iken, bu iki saldırıda 34 ve 81'e yükselmiştir. Taşma saldırılarında, hedef düğüm oranının ve hedef düğüm süresinin ortalaması diğerlerine göre fazladır. Sürüm numarası değiştirme saldırısında, toplam paket süresinin ortalaması diğerlerine göre düşük kalmıştır. Taşma saldırılarında, toplam paket uzunluğunun ortalaması 5713,654 'tür iken diğer ortalamalara ait veriler 1900-2900 bandındadır. Yine taşma saldırılarında, hedef paket oranı diğer ortalamalara göre yüksek kalmıştır. Tüm saldırılarda DIO mesaj ortalamasında bir artış tespit edilmiştir. DIS, DAO ve diğer mesaj sayıları ortalamalarında, yine taşma saldırıları diğer ortalamalara göre anormal gözükmektedir. Bunun nedeni, taşma saldırılarında zafiyetli düğümün sürekli DIS mesajları iletmesidir. Gönderilen DIS mesajlarının diğer düğümler tarafından kabul edilmesi de diğer mesaj ortalamasını artırmıştır. Yapılan simülasyonda da bariz bir şekilde bu fark gözükmektedir.

#### 3.4.2.2 Medyan Değerlerinin İncelenmesi

Taşma saldırılarında ve sürüm numarası artırma saldırılarında, paket sayıları medyanlarında diğer medyanlara göre yüksektir. Taşma saldırılarında ve sürüm numarası artırma saldırılarında kaynak düğüm oranı, kaynak düğüm süresi, kaynak paket oranı medyanları diğer medyanlara göre düşük, toplam paket süresi, toplam paket uzunluğu medyanları diğer medyanlara nazaran yüksektir. Sürüm numarası artırma saldırısında, DIO sayısı medyanı diğerlerinden büyük oranda ayrışmıştır. Diğerleri, 0,5 ve 7 iken DIO sayısı medyanı 26'dır. Bunun nedeni, sürüm numarası güncellendiğinde DODAG yapısının yeniden oluşması ve düğümlerin daha fazla sayıda DIO mesajları iletmesidir. Sürüm numarası artırma saldırısında, DIO sayısı medyanındaki artış anomalinin tespitinde önemli yeri haizdir.

#### 3.4.2.3 Maksimum Değerlerinin İncelenmesi

Tüm saldırı türlerinde, toplam paket sürelerinin maksimum değerleri diğer maksimum değerlerinden ayrışmaktadır. Azaltılmış rank saldırılarında, toplam paket süresi maksimum değeri saldırı olmayan değerlere göre aşırı büyükken, taşma ve sürüm numarası artırma saldırılarında bu değer diğer saldırı olmayan değerlere göre aşırı

küçüktür. Tüm saldırı tüplerinde, DIO sayısı maksimum değerleri oldukça yüksektir. Normal düğümlerle yapılan simülasyonlarda, 30-63 bandında olan bu değerler, tüm saldırılarda 197-214 bandındadır. Taşma saldırılarında zafiyetli düğümün sürekli DIS mesajları ilettiği için DIS mesajlarının anormal bir şekilde oluştuğu gözükmektedir. Diğer veri setlerinde hiç DIS mesajı yokken, taşma saldırılarında anormal bir artış olduğu gözükmektedir. Diğer düğümler de gelen DIS mesajlarına cevap verdiği için diğer mesaj sayılarındaki bariz artış, maksimum değerinin aşırı fazla olmasına neden olmuştur.

#### 3.4.2.4 Minimum Değerlerinin İncelenmesi

Taşma saldırılarında hedef güğüm oranı minimum değeri diğerlerine göre ayrışmıştır.

#### 3.4.2.5 Standart Sapma Değerlerinin İncelenmesi

Taşma saldırılarında ve sürüm numarası artırma saldırılarında paket sayısı, toplam paket uzunluğu standart sapma değerleri diğer değerlere göre aşırı yüksek, toplam paket süresi standart sapma değerleri diğer değerlere göre aşırı düşüktür. Azaltılmış Rank saldırılarında DIO sayısı standart sapma değeri, diğer değerlere göre oldukça yüksek kalmıştır. Taşma saldırılarında, zafiyetli düğüm sürekli DIS mesajı gönderdiği için, standart sapma değeri de oluşmuştur. Sürüm Numarası Artırma Saldırılarında, DAO sayısı standart sapma değeri diğerlerine göre yüksektir.

#### 3.4.2.6 Çarpıklık Değerlerinin İncelenmesi

Taşma saldırılarına ait paket sayısı, hedef düğüm oranı, hedef düğüm süresi, toplam paket süresi, hedef paket oranı çarpıklığı, diğer verilere göre oldukça küçüktür. Tüm saldırılarda DIO sayısı çarpıklığı, normal düğümle yapılan simülasyon verilerine göre yüksek görülmüştür. Taşma saldırılarında, zafiyetli düğüm sürekli DIS mesajı gönderdiği için, çarpıklık değeri de oluşmuştur.

#### 3.4.2.6 Basıklık Değerlerinin İncelenmesi

Taşma saldırılarındaki paket sayısı basıklığı diğer verilerin basıklığına göre oldukça düşüktür. Azaltılmış rank saldırılarında, toplam paket süresi ve DIO sayısı basıklığı diğer basıklık verilerine göre aşırı yüksektir. Taşma saldırılarında DAO mesaj sayısı basıklığı diğer basıklık verilerine göre aşırı yüksektir.

#### 3.4.2.7 İstatistik Verilerinin Genel Değerlendirmesi

Tüm istatistiksel veriler göz önünde alındığında, taşma saldırılarında ciddi bir anomali söz konusudur. Bu anomali de makine öğrenmesi algoritmaları ile yüksek oranda tespit edilmesi ile sonuçlanmıştır. Özellikle bu saldırıda DIS mesajlarının oluşması normal simülasyon verilerine göre ciddi bir fark oluşturmuştur.

Sürüm Numarası artırma saldırıları istatistiksel verilerinin normal düğümlerle yapılan simülasyon verilerine göre de farklılaştığı gözlemlenmiştir.

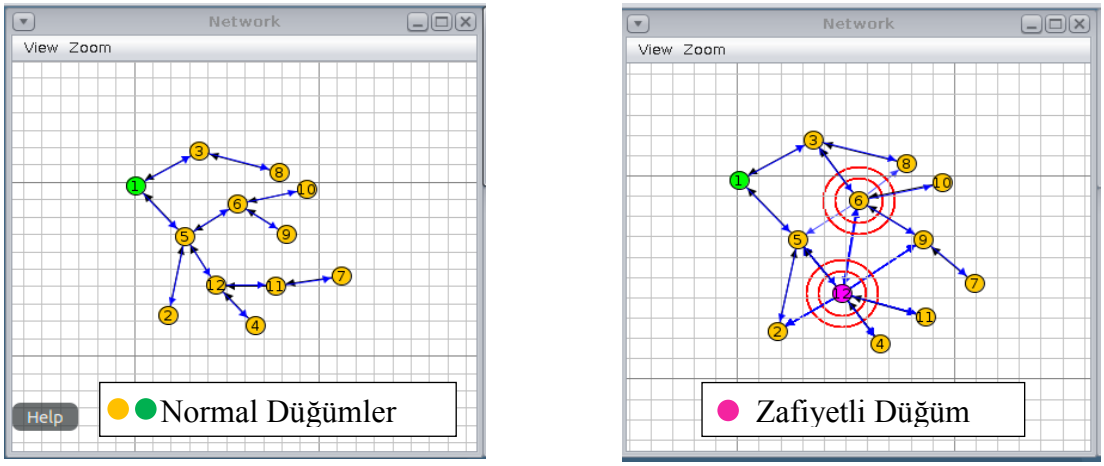
Azaltılmış rank saldırılarının tespiti için en bariz şekilde gözlemlenen istatistiksel veri toplam paket süresi ve DIO sayısı basıklığıdır.

### 3.4.3 Makine Öğrenmesi Değerlerinin Yorumlanması

#### 3.4.3.1 Taşma Saldırısına Ait Değerlerin Yorumlanması.

Taşma saldırısı veya diğer bir adıyla "Hello Flood" saldırısına ait oluşturulan veri seti toplam 428 satırdır. Aynı veri seti, farklı makine öğrenmesi algoritmaları ile eğitildiğinde ve aynı değerler teste tabi tutulduğunda en fazla doğruluk oranına sahip algoritma %97,2 doğruluk oranı ile **Derin Öğrenme** algoritmasıdır. Bu algoritmayı sırasıyla %95,7 doğruluk oranı ile Lojistik Regresyon ve K En Yakın Komşu, %95 doğruluk oranı ile Rastgele Orman, %93,6 doğruluk oranı ile Karar Ağaçları, %69,7 doğruluk oranı ile Navie Bayes algoritmaları izlemektedir. Eğitim süreleri göz önüne alındığında, **Karar Ağaçları** sınıflandırma algoritması 1 milisaniyenin altında işlemi gerçekleştirmiştir. Bu algoritmayı sırasıyla 4 milisaniye ile K En Yakın Komşu, 168 milisaniye ile Rastgele Orman, 363 milisaniye ile Lojistik Regresyon ve 1847 milisaniye ile Yapay Sinir Ağları takip etmektedir.

Taşma saldırılarında ağ paketleri anormal davranışlar sergiler. Taşma saldırıları büyük miktarda trafik oluşturur ve düğümleri ve bağlantıları kullanılamaz hale getirir. Bu saldırı zafiyetli bir düğümün gereksiz DIS mesajları göndermesi ile oluşur. Bu farklılık şekil 3.10'da açıkça görülmektedir. Saldırı olan bir ağın paketlerindeki DIS oranı normal bir ağdaki DIS oranları ile farklılık gösterir. Bu nedenle bu saldırı kolaylıkla tespit edilebilmiştir. Ayrıca istatistiksel metotların karşılaştırılması bölümünde de anomali fark edilmektedir.




**Şekil 3.10: Normal Düğümler ve Zafiyetli Düğümle Yapılan Simülasyon.**

Yapay sinir ağları, gelişmiş bir algoritmadır. Bu algoritma, Taşma Saldırıları için oluşturulmuş elimizdeki veri setindeki anomaliyi yüksek doğrulukla tespit etmiştir.

DIS oranlarının farklılık göstermesi, sonucun olup olmamasını doğrudan etkilediği için etkiliyorsa doğrusal bir denklemle ifade edilebilir. Bu yüzden Lojistik Regresyon algoritması da %95'in üzerinde bir sonuç vermiştir.

Taşma saldırılarında, veri paketlerindeki "kategorik" veri niteliğindeki DIS oranlarının farklılık göstermesi, özellikle Karar Ağaçları algoritmalarının bu saldırıyı kolay tespit edebilmesini sağlar. Çünkü DIS oranlarındaki "bilgi kazancı" diğer girdilere göre daha yüksektir. Eğitim süresinin kısa olması, hızlı çalışmasından kaynaklanır.

KNN algoritmasının, yüksek doğrulukla saldırıyı tespit edebilmesinin nedeni, yine DIS mesajlarının anomaliliğinden kaynaklanır.

### 3.4.3.2 Sürüm Numarası Artırma Saldırısına Ait Değerlerin Yorumlanması

Sürüm Numarası Artırma Saldırısına ait oluşturulan veri seti toplam 336 satırdır. Aynı veri seti, farklı makine öğrenmesi algoritmaları ile eğitildiğinde ve aynı değerler teste tabi tutulduğunda en fazla doğruluk oranına sahip algoritma %81 doğruluk oranı ile **K En Yakın Komşu** algoritmasıdır. Bu algoritmayı sırası ile %74,8 doğruluk oranı ile Lojistik Regresyon, Karar Ağaçları, Navie Bayes, %72,9 doğruluk oranı ile Rastgele Orman, %72 doğruluk oranı ile Yapay Sinir Ağları izlemektedir. Eğitim süreleri göz önüne alındığında 2 milisaniye ile **KNN ve Karar Ağaçları** makine öğrenmesi algoritmaları en kısa sürede eğitilmiştir. Bu algoritmayı, 5 milisaniye Navie Bayes, 185 milisaniye ile Lojistik Regresyon, 253 milisaniye ile Rastgele Orman, 1688 milisaniye ile Yapay Sinir Ağları takip etmektedir.

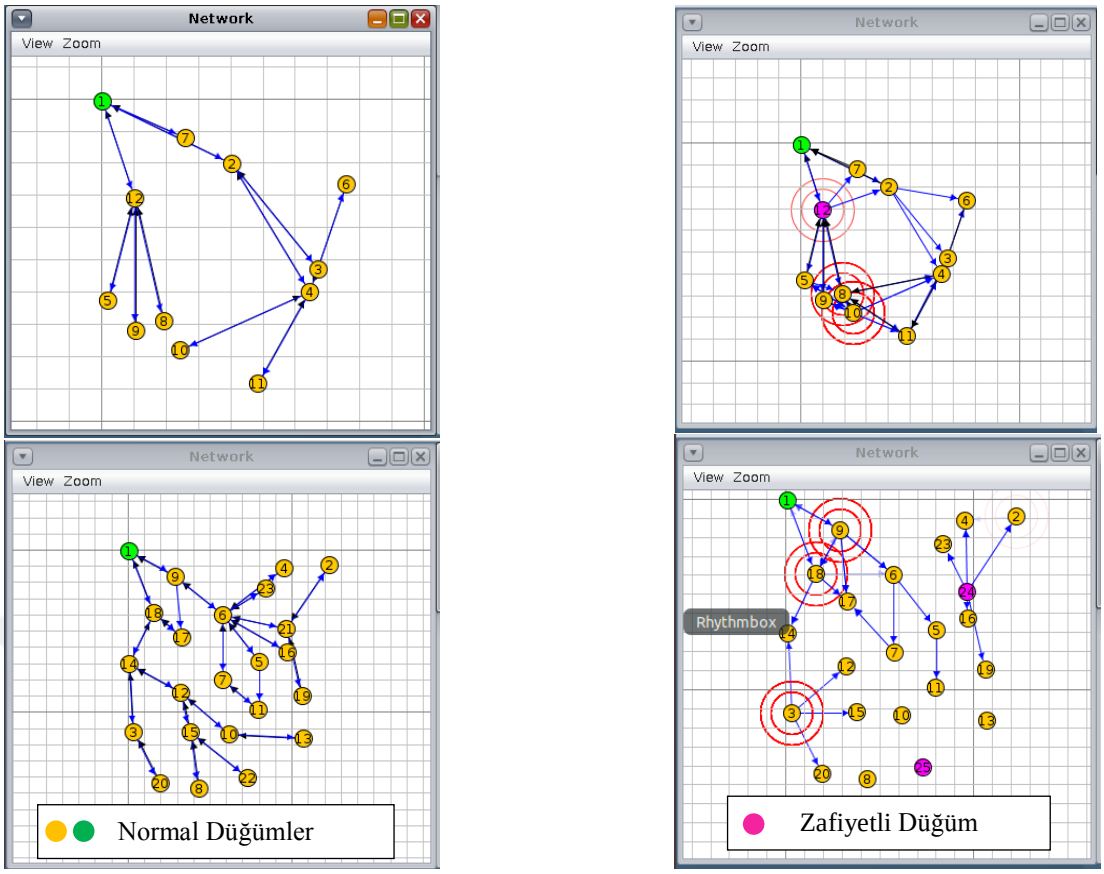
Versiyon Numarası artırma Saldırısı, zafiyetli bir düğümün, versiyon numarasını değiştirip komşularına iletmesi ile gerçekleşir. Bu gerçekleştiğinde, diğer düğümler normal DODAG yapısının değiştiğini anlar ve DODAG yeniden oluşturulmak istenir. DODAG'ın art arda gereksiz yeniden yapılandırılması, mesaj ek yükünü önemli ölçüde artırır, düğüm kaynaklarını tüketir ve ağı tıkar. Versiyon Numarası Artırma Saldırısına ait görüntüler Şekil-3.11''de yer almaktadır. Özellikle 22 düğümle yapılan deneyde, DODAG yapısının oluşmadığı Şekil-3.11'de görülmektedir.

DODAG'ın her defasında yeniden oluşması DAO, DIA ve DAO-ACK mesajlarının ve veri paketi uzunluklarının artmasına neden olur.

KNN algoritmasının, yüksek doğrulukla saldırıyı tespit edebilmesinin nedeni, yine DAO, DIA ve DAO-ACK mesajlarının ve veri paketi uzunluklarının anomaliliğinden kaynaklanır.

DAO, DIA ve DAO-ACK mesajlarının ve veri paketi uzunluklarının normal düğümlerle yapılan simülasyondan elde edilen değerler göre farklılık göstermesi, yine doğrusal bir denklemle ifade edilebilmeyi kolaylaştırdığı için Lojistik Regresyon algoritması da saldırıyı yespit edebimiştir.

Yapay sinir ağları, gelişmiş bir algoritmadır. Bu algoritma da, bu saldırı için oluşturulmuş elimizdeki veri setindeki anomaliyi yüksek doğrulukla tespit edebilir.




**Şekil 3.11: 11 ve 23 Düğümle Yapılan Sürüm Numarası Saldırısı Görüntüleri.**

### 3.4.3.3 Azaltılmış Rank Saldırısına Ait Değerlerin Yorumlanması

Azaltılmış Rank Saldırısına ait oluşturulan veri seti toplam 302 satırdır. Aynı veri seti, farklı makine öğrenmesi algoritmaları ile eğitildiğinde ve aynı değerler teste tabi tutulduğunda en fazla doğruluk oranına sahip algoritma %58 doğruluk oranı ile **Rastgele Orman** ve **Yapay Sinir Ağları** algoritmasıdır. Bu algoritmayı sırası ile %57 doğruluk oranı ile **Karar Ağaçları**, %56 doğruluk oranı ile **K En Yakın Komşu**, %54 doğruluk oranı ile Lojistik Regresyon ve Navie Bayes algoritmaları izlemektedir. Eğitim süreleri göz önüne alındığında **K En Yakın Komşu** makine öğrenmesi algoritması 1 milisaniyenin altında eğitilmiştir. Bu algoritmayı, 1 milisaniye ile Navie Bayes, 2 milisaniye ile Karar Ağaçları, 5 milisaniye ile Lojistik Regresyon, 10 milisaniye ile Rastgele Orman ve 1720 milisaniye ile Yapay Sinir Ağları takip etmektedir.

Diğer saldırılarla karşılaştırıldığında, Azaltılmış Rank saldırısının tespit edilme oranı oldukça düşüktür. Bunun nedeni, bu saldırının amacının hizmeti engellemesi veya sistemin çalışmasını aksatması olmamasıdır. Azaltılmış rank saldırısı, gizliliği ihlal

eden saldırılardandır. Bu saldırıda, zafiyetli düğüm, anormal bir şekilde daha düşük bir rank duyurduğunda, birçok normal düğüm saldırgan aracılığıyla DODAG grafiğine bağlanır. Bu saldırı ağa zarar vermediği için tespit edilmesi diğerlerine nispeten daha zordur çünkü normal ve zafiyetli veri paketleri benzer özellikler gösterir.

# 4. SONUÇ

Bu tez çalışması ile, RPL protokolünde gerçekleşecek taşma, sürüm numarası artırma ve azaltılmış rank saldırılarını tespit edebilmek maksadıyla IoT cihazları, cooja simülatöründe tamamen normal düğümlerle ve içinde zafiyetli düğüm barındıran normal düğümlerle simüle edilmiş, simülasyon sonucunda ham veri seti oluşturulmuş, bu ham veri seti anlamlı hale getirilerek, zafiyetli ve normal olarak etiketlenmiştir. Anlamlı hale getirilen yeni veri seti, 1/3 oranında test ve 2/3 oranında eğitim veri seti olarak ayırılmış, eğitim veri seti "Karar Ağacı", "Lojistik Regresyon", "Rasgele Orman", "Naive Bayes", "K En Yakın Komşu" ve "Yapay Sinir Ağları" algoritmaları ile eğitilmiş ve test edilmiştir. Saldırılara ilişkin test sonuçları karşılaştırılmış, karşılaştırma sonucunda, Taşma Saldırılarının tespitinde **%97,2** doğruluk oranı ile **Yapay Sinir Ağları** algoritması, Sürüm Numarası Artırma Saldırılarının tespitinde **%81** doğruluk oranı ile **K En Yakın Komşu** algoritması, Azaltılmış Rank saldırılarının tespitinde **%58** doğruluk oranı ile **Yapay Sinir Ağları** algoritması başarı gösterdiği tespit edilmiştir.

Bu tez çalışması ile özellikle Taşma ve Sürüm Numarası yükseltme saldırılarını tespit edebilecek hızlı, güvenilir, basit bir model oluşturulmuş, bu model ile en iyi sonucu verebilecek makine öğrenmesi algoritması belirlenmiştir. Deneyler sonucunda, verilerin anlamlı hale getirilmesi aşaması doğrudan ağ paketleri analiz edilerek yapılmıştır.

(Yavuz, 2018), ham verileri sunduğu yöntemle "zenginleştirdikten" sonra sadece derin öğrenme algoritması ile Azaltılmış Rank Saldırılarını 94.7%, Taşma Saldırılarını 99%, Sürüm Numarası Artırma Saldırılarını 95% doğruluk oranı ile tespit etmiştir. Ancak deneyleri cooja simülatöründe 100-1000 arasında değişen düğümle gerçekleştirmiştir. 250 düğümden fazla çalıştırılan simülatör, hem işlemciyi aşırı yormakta hem de bellek kaynaklarını çok tüketmektedir. Bu nedenlerden dolayı simülatörün verilerin doğruluğunun yeniden gözden geçirilmesi gerektiği düşünülmektedir. Dolayısı ile bu çalışmadan elde edilen sonuç daha az düğümle yeniden sonuçlandırılmalıdır.

(Müller, Debus, Kowatsch, & Böttinger, 2019), RPL'de gerçekleşen, Azaltılmış Rank Saldırısını %68, Taşma Saldırısını 90%, Sürüm Numarası Artırma Saldırısını, 96% doğruluk oranı ile tespit etmişlerdir. Bu tez çalışmasında sunulan teknik ile azaltılmış rank saldırısı aynı nispette tespit edilirken, taşma saldırıları daha iyi bir performansla tespit edilmiştir. Müller ve arkadaşlarının sunduğu çözüm sürüm numarası artırma saldırılarının tespitinde daha başarılı olmuştur.

(Verma & Ranga, 2019), Sinkhole, Kara Delik, Sybil, Clone ID, Selective Forwarding, Hello Flooding and Local Repair saldırılarını geliştirdikleri ELNIDS isimli yöntemle %94.5 oranında tespit edebilmişlerdir. Ancak çalışmalarında kullandıkları veri setinin 20 adet özelliği barındırdığından bahsedilmekte, bu 20 özelliğin neler olduğundan bahsedilmemektedir. Bu tez çalışmasında sadece ağ paketleri verilerinden oluşan veri setinden yararlanılmıştır.

(Neerugatti & Reddy, 2019), rank saldırılarını KNN tabanlı bir makine öğrenmesi algoritması olan MLTKNN algoritması ile farklı sayıda düğümler ile (5-30) %90 ve üzerinde doğruluk oranı ile saldırıyı tespit etmişlerdir. Ancak ilgili makalede bahsedilen rank saldırısının artırılmış rank saldırısı mı yoksa azaltılmış rank saldırısı mı olduğu bilinmemektedir.

Belavagi (2020) ve Çakır (2020) yaptıkları çalışmalarda ağ paketleri dışında başka parametreler de kullanmışlardır. (Güç Tüketimi, İşlemci yoğunluğu gibi) Pratikte bu değerleri almak ve işlemek ağda ayrı bir maliyet gerektirecektir. Ancak, bu tez çalışmasında sunulan veri anlamlandırılması ve uygun makine öğrenmesi metotları ile daha az maliyetle hızlı ve etkili bir çözüm bulunmuştur. Aynı zamanda, tek bir makine öğrenmesi ile saldırıların tespiti gerçekleştirilmemiş, yine hızlı ve güvenilir algoritmalar karşılaştırılarak en uygun makine öğrenmesi metodu saptanmıştır.

Taşma, Sürüm Numarası Artırma ve Azaltılmış Rank saldırılarını tespit etmek maksadıyla ağda oluşan paketler doğrudan bu tez çalışmasında sunulan veri anlamlandırılması metodu ile özetlendikten sonra **Rastgele Orman** algoritması ile eğitilip kullanılabilir. Rastgele Orman algoritması üç saldırıyı da yüksek doğrulukla tespit edebilmektedir.

Azaltılmış Rank saldırılarının sunulan yöntem ile tespit edilebilmesi diğer saldırılara nispetle oldukça düşüktür. (Seth, Biswas, & Dhar, 2020) Gidiş-Dönüş Sürelerini kullanarak Azaltılmış Rank Saldırılarını tespit etmek için bir model geliştirmiştir.

Ancak, bu çalışma 3. katman ağ paketlerinin özetlenmesi ile yapılmamıştır. Tablo 3.5 yeniden incelendiğinde, Azaltılmış Rank saldırısında oluşan network paketleri, veri anlamlandırması sürecinden sonra diğer saldırılara nazaran çok daha fazla satırla özetlenmiştir. Bu da anormal bir durumun olduğunun işaretidir, ancak makine öğrenmesi algoritmaları, mevcut modelle bu saldırıyı yüksek doğruluk oranı ile tespit edememiştir. Gelecekte, Azaltılmış Rank saldırısını 3'üncü katman ağ paketlerini işleyerek tespit edebilen bir model geliştirilebileceği değerlendirilmektedir.

# KAYNAKÇA

(ETSI), E. T. (2012). *www.portal.etsi.org*. https://portal.etsi.org/Services/Centre-for-Testing-Interoperability/Activities/M2M/oneM2M adresinden alındı

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials, 17*(4), 2347-2376. doi:10.1109/COMST.2015.2444095

Al-Garadi, M. A., Mohamed, A., Al-Ali, A., Du, X., Ali,, I., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys & Tutorials*, 1-44.

Banks, A., & Gupta, R. (28 Eylül 2014). *MQTT Version 3.1.1*. Candidate OASIS Standard 02. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. adresinden alındı

Belavagi, M. C., & Muniyal, B. (2020). *Multiple intrusion detection in RPL based networks.* International Journal of Electrical and Computer Engineering (IJECE). doi:10.11591/ijece.v10i1.pp467-476

Bhunia, S. S., & Gurusamy, M. (2017). Dynamic Attack Detection and Mitigation in IoT using SDN. *27th International Telecommunication Networks and Applications Conference (ITNAC).*

Bilgin, M., & Yılmaz, A. (2018). *Makine Öğrenmesi.* İstanbul: Papatya Bilim.

Bonaccorso, G. (2017). *Machine Learning Algorithms.* Birmingham: Packt Publishing Ltd.

Burhan, M., Rehman, R. A., Khan, B., & Kim, B.-S. (2018). IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey. *Sensors*, 1-37. doi:10.3390/s18092796

Canedo, J., & Skjellum, A. (2016). Using Machine Learning to Secure IoT Systems.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2018). Network Intrusion Detection for IoT Security based on Learning Techniques. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, 1-32.

Cheshire, S., & Krochmal, M. (Şubat 2013). *DNS-Based Service Discovery.* Internet Engineering Task Force (IETF).

Cheshire, S., & Krochmal, M. (Şubat 2013). *Multicast DNS.* Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc6762 adresinden alındı

Chiu, E., Lin, J., Mcferron, B., Petigara, N., & Seshasai, S. (2001). *Mathematical Theory of Claude Shannon.* The Structure of Engineering Revolutions.

Cuomo, F., Della Luna, S., Cipollone, E., Todorova, P., & Suihko, T. (2008). Topology Formation in IEEE 802.15.4: Cluster-Tree Characterization. *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 276-281. doi:10.1109/PERCOM.2008.26

Çakır, S., Toklu, S., & Yalçın, N. (2020). *RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning.* EEE Access. doi:10.1109/ACCESS.2020.3029191

Deogirikar, J., & Vidhate, A. (2017). Security Attacks inIoT: A Survey. *International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, (s. 32-37). India.

D'Hondt, A., Bahmad, H., Vanhee, J., & Sadre, R. (2015). *RPL attacks framework.* Louvain-la-Neuve, Belgium: Universit catholique de Louvain. https://github.com/dhondta/rpl-attacks adresinden alındı

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. *2018 IEEE Symposium on Security and Privacy Workshops*, 29-35. doi:DOI 10.1109/SPW.2018.00013

Grasso, J. (2004). The EPCglobal network: overview of design, benefits, & security. *EPCglobal*, 24.

Gutierrez, J., Naeve, M., Callaway, E., Bourgeois, M., Mitter, V., & Heile, B. (2001). IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks. *IEEE Network*, 12-19. doi:10.1109/65.953229

HASSIJA, V., CHAMOLA, V., SAXENA, V., JAIN, D., GOYAL, P., & SIKDAR, B. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access*, 82721-82743.

Howitt, I., & Gutierrez, J. (2003). IEEE 802.15.4 low rate - wireless personal area network coexistence issues. *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, 1481-1486 vol.3. doi:10.1109/WCNC.2003.1200605

Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Communications Surveys & Tutorials*, 1-38.

Infosec. (2016, 10 29). *www.infosec.com.tr*. https://www.infosec.com.tr/dyne-karsi-yapilan-ddos-saldirisi/ adresinden alındı

Iqbal, M. A., Hussain, S., Xing, H., & Imran, M. A. (2021). IoT Protocol Stack. *Enabling the Internet of Things: Fundamentals, Design and Applications, IEEE*, 93-126. doi:10.1002/9781119701460.ch5.

KURİŞ, U. (2020). *Nesnelerin İnterneti Ekosisteminde Yapay Zeka Tabanlı Saldırı Tespit Sistemi Geliştirilmesi.* Yüksek Lisans Tezi, İstanbul.

Le, A., Loo, J., Luo, Y., & Lasebae, A. (2011). Specification-based IDS for securing RPL from topology attacks. *2011 IFIP Wireless Days (WD)*, 1-3. doi:10.1109/WD.2011.6098218

Lee, S.-Y., Wi, S.-r., Seo, E., Jung, J.-K., & Chung, T.-M. (2017). ProFiOt: Abnormal Behavior Profiling (ABP) of IoT devices based on a Machine Learning Approach. *27th International Telecommunication Networks and Applications Conference (ITNAC).*

Li, F., Shinde, A., Shi, Y., Ye, J., Li, X.-Y., & Song, W. (2019). System Statistics Learning-Based IoT Security: Feasibility and Suitability. *IEEE INTERNET OF THINGS JOURNAL, VOL. X, NO. X, XX 2019*. doi:DOI 10.1109/JIOT.2019.2897063

Lueth, K. L. (2014, Aralık 19). *iot-analytics*. 2021 tarihinde https://iot-analytics.com/internet-of-things-definition/ adresinden alındı

Mackensen, E., & Lai, M. (2012). Bluetooth Low Energy (BLE) based wireless sensors. *SENSORS, 2012 IEEE*, 1-4. doi:10.1109/ICSENS.2012.6411303

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015, Ocak). *Journal of Computer and Communications*. https://www.scirp.org/html/56616_56616.htm adresinden alındı

Mayzaud, A., Badonnel, R., & Chrisment, I. (2016). A Taxonomy of Attacks in RPL-based Internet. *International Journal of Network Security, ACEEE a Division of Engineers Network*, 459 - 473. https://hal.inria.fr/hal-01207859/document adresinden alındı

Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., & Elovici, Y. (2017). Detection of Unauthorized IoT Devices Using Machine Learning Techniques. *arXiv:1709.04647v1 [cs.CR] 14 Sep 2017*.

Moh, M., & Raju, R. (2018). Machine Learning Techniques for Security of Internet of Things (IoT) and Fog Computing Systems. *2018 International Conference on High Performance Computing & Simulation* (s. 709-715). USA: IEEE. doi:DOI 10.1109/HPCS.2018.00116

Mrabet, H., Belguith, S., Alhomoud, A., & Jemai, A. (2020). A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *www.mdpi.com/journal/sensors*, 1-19.

Müller, N., Debus, P., Kowatsch, D., & Böttinger, K. (2019). *Distributed Anomaly Detection of Single Mote Attacks in RPL Networks.* Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (ICETE 2019),. doi:DOI: 10.5220/0007836003780385

Neerugatti, V., & Reddy, A. M. (2019). *Machine Learning Based Technique for Detection of Rank Attack in RPL based Internet of Things Networks.* International Journal of Innovative Technology and Exploring Engineering.

(2012). *OASIS Advanced Message Queuing Protocol (AMQP).* http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html adresinden alındı

Olyaei, B., Pirskanen, J., Raeesi, O., Hazmi, A., & Valkama, M. (2013). Performance comparison between slotted IEEE 802.15.4 and IEEE 802.1 lah in IoT based applications. *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 332-337. doi:10.1109/WiMOB.2013.6673381

P. Saint, A. (2004). *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence.* Jabber Software Foundation. https://www.hjp.at/doc/rfc/rfc3922.html adresinden alındı

Pardo- Castellote, G. P., Farabaugh , B., & Warren, R. (2005). *An Introduction to DDS and Data-Centric Communications.* Real- Time Innovations.

Project, B. (2012, Temmuz 11). *www.iot-butler.eu*. https://www.slideshare.net/butler-iot/butler-project-overview-13603599 adresinden alındı

Rahman, A., & Dijk, E. (2014, Ekim). *Internet Engineering Task Force (IETF)-7390*. https://tools.ietf.org/html/rfc7390 adresinden alındı

Raj, P., & Raman, A. (2017). *The Internet of Things: Enabling Technologies, Platforms, and Use Cases.* New York: CRC Press.

Rentz, P. (2019, Ocak 17). *OWASP Releases Latest Top 10 IoT Vulnerabilities*. Aralık 23, 2020 tarihinde https://www.techwell.com/techwell-insights/2019/01/owasp-releases-latest-top-10-iot-vulnerabilities adresinden alındı

Restuccia, F., D'Oro, S., & Melodia, T. (2018). Securing the Internet of Things in the Age of Machine Learning and Software-defined Networking. *IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 1, JANUARY 2018*, 1-14.

Roldán, J., Boubeta-Puig, J., Martínez, J. L., & Ortiz, G. (2020). Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. *www.elsevier.com/locate/eswa*, 1-21.

Roopak, M., Tian, G. Y., & Chambers, J. (2019). Deep Learning Models for Cyber Security in IoT Networks. *978-1-7281-0554-3/19/$31.00©2019 IEEE*, 0452-0457.

Roukounaki, A., Efremidis, S., Soldatos, J., Neises, J., Walloschke, T., & Kefalakis, N. (2018). Scalable and Configurable End-to-End Collection and Analysis of IoT Security Data.

Safaric, S., & Malaric, K. (2006). ZigBee Wireless Standard. *Proceedings ELMAR 2006*, (s. 259-262). Zadar, Croatia. doi:10.1109/ELMAR.2006.329562

Seth, A. D., Biswas, S., & Dhar, A. K. (2020). Detection and Verification of Decreased Rank Attack using Round-Trip Times in RPL-Based 6LoWPAN Networks. *2020*

*IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 1-6. doi:10.1109/ANTS50601.2020.9342754.

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems 107 (2020)*, 433-442.

Shelby, Z., & Bormann, C. (2009). *6LoWPAN: The Wireless Embedded Internet.* UK: WILEY.

Sivanathan, A., Gharakheili, H. H., & Sivaraman, V. (2020). Managing IoT Cyber-Security Using Programmable Telemetry and Machine Learning. *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 17, NO. 1*,, 60-74.

Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. *www.elsevier.com/locate/jnca*, 1-18.

Verma, A., & Ranga, V. (2019). *ELNIDS: Ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things.* 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). doi:10.1109/IoT-SIU.2019.8777504

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., . . . Alexander, R. (Mart 2012). *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.* Internet Engineering Task Force. https://www.hjp.at/doc/rfc/rfc6550.html adresinden alındı

Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT Security Techniques Based on Machine Learning. *IEEE Signal Processing Magazine*, 42-49.

Yavuz, F. Y. (2018). Deep Learning in Cyber Security for Internet of Things.

Yılmaz, A. (2015). *Sinirsel Bulanık Mantık Modeliyle Kanser Risk Analizi.* Sakarya: Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Zeadally, S., & Tsikerdekis, M. (2019). Securing Internet of Things (IoT) with machine learning. *wileyonlinelibrary.com/journal/dac*, 1-16. doi:DOI: 10.1002/dac.4169

Zhang, N., Demetrio, S., Mi, X., Diao, W., Yuan, K., Zong, P., . . . Lin, Y.-H. (2017). Understanding IoT Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be. *arXiv:1703.09809v1 [cs.CR] 28 Mar 2017*.

Zheng, J., & Lee, M. (2006). A comprehensive performance study of IEEE 802.15. 4. *Sensor Network Operations 4*, 218–237.

ZigBee. (2020). *ZigBee*. https://zigbeealliance.org/ adresinden alındı

## EKLER

### EK 1 : Simülatör Tarafından Oluşturulan JSON Dosyasının İçeriği


```
{
 "BASE": {
  "simulation": {
   "number-motes": 20,
   "target": "z1",
   "duration": 120,
   "area-square-side":500}},
 "hello-flood": {
  "simulation": {
   "title": "Test hello-flood simulation",
   "goal": "Create a new simulation",
   "root": "echo"},
  "malicious": {
   "type": "sensor",
   "building-blocks": [
    "hello-flood"
   ]}},
 "increased-version": {
  "simulation": {
   "title": "Test increased-version simulation",
   "goal": "Create a new simulation",
   "root": "echo"
  },
  "malicious": {
   "type": "sensor",
   "building-blocks": [
    "increased-version"
   ]
  }
 },
 "decreased-rank": {
  "simulation": {
   "title": "Test decreased-rank simulation",
   "goal": "Create a new simulation",
   "root": "echo"
  },
  "malicious": {
   "type": "sensor",
   "building-blocks": [
    "decreased-rank" ] }  } }
```

**EK 2: Zafiyetli Düğüm Oluşturmak İçin Eklenen Kod Blokları**

```
{
  "hello-flood": {
    "RPL_CONF_DIS_INTERVAL": 0,
    "RPL_CONF_DIS_START_DELAY": 0,
    "rpl-timers.c": ["next_dis++;", "next_dis++; int i=0; while (i<20) {i++;
dis_output(NULL);}"]
  },
  "increased-version": {
    "rpl-icmp6.c": ["dag->version;", "dag->version++;"]
  },
  "decreased-rank": {
    "RPL_CONF_MIN_HOPRANKINC": 0,
    "rpl-private.h": [
      ["#define RPL_MAX_RANKINC            (7 * RPL_MIN_HOPRANKINC)",
"#define RPL_MAX_RANKINC 0"],
      ["#define INFINITE_RANK              0xffff", "#define INFINITE_RANK
256"]
    ],
    "rpl-timers.c": ["rpl_recalculate_ranks();", null]
  }
}
```

- Taşma Saldırılarını gerçekleştirmek için gereksiz DIS mesajları yollanmıştır.
- Sürüm Numarası Artırma Saldırılarında, sürüm numarasını +1 artırmıştır.
- Azaltılmış Rank saldırılarında, cihazlar rank numaralarını azaltmışlardır.

**EK 3:Veri Setinin Anlamlandırılması İçin Oluşturulan Python Kodları**

```
import pandas as pd
import numpy as np

flnm="VN-10N1R.csv"
filestr="Attack Files/CSV/"+flnm
if "M" in flnm:
   lbl = 1
else:
   lbl = 0
resultfile="Results/New/"+flnm
Raw_Data=pd.read_csv(filestr,index_col="No.")
np_Raw_Data=np.array(Raw_Data)
#np_Raw_Data=np.delete(np_Raw_Data,0,axis=0)
np_Raw_Data=np_Raw_Data[np.argsort(np_Raw_Data[:, 0])]

packetDurations=[]
counter=0
while counter < len(np_Raw_Data):
   duration=0
   if counter != 0 and counter+1<len(np_Raw_Data):
      duration = np.float32(np_Raw_Data[counter][0])-
np.float32(np_Raw_Data[counter - 1][0])
   packetDurations.append(duration)
   counter+=1

packetDurations=np.delete(packetDurations,0,axis=0)
np_Raw_Data=np.delete(np_Raw_Data,len(np_Raw_Data)-1,axis=0)

np_Raw_Data=np.insert(np_Raw_Data,1,packetDurations,axis=1)

source_unique_array=np.unique(np.array(Raw_Data.iloc[:,1:2].astype(str)))
destination_unique_array=np.unique(np.array(Raw_Data.iloc[:,2:3].astype(str)))
info_unique_array=np.unique(np.array(Raw_Data.iloc[:,5:6]))
protocol_unique_array=np.unique(np.array(Raw_Data.iloc[:,3:4]))

all_ip_addresses=np.concatenate((source_unique_array,destination_unique_array))

all_ip_addresses=np.unique(all_ip_addresses)

count=1
ip_dict={}

from sklearn import preprocessing
le=preprocessing.LabelEncoder()
lb_all_ip_addresses=le.fit_transform(all_ip_addresses)

cnt=0
for x in all_ip_addresses:
   ip_dict[x]=lb_all_ip_addresses[cnt]
```

```
    cnt+=1

np_Raw_Data=np_Raw_Data[np.argsort(np_Raw_Data[:, 0])]

duration=np.floor(np.float32(np_Raw_Data[-1][0]))
counter=0
currentSecond = 60.0
packetcount={}
TotalPacketDuration={}
TotalPacketLenght={}
src_count={}
dst_count={}
src_duration={}
dst_duration={}
src_packet_lenght_sum={}
dst_packet_lenght_sum={}
DioCount={}
DisCount={}
DaoCount={}
OtherMsg={}
frame=[]
row=pd.DataFrame(columns=['second','src',                    'dst','packetcount','src_ratio',
'dst_ratio','src_duration_ratio',
'dst_duration_ratio','TotalPacketDuration','TotalPacketLenght','src_packet_ratio','dst_
packet_ratio','DioCount','DisCount','DaoCount','OtherMsg','label'])
while counter<duration:
    #one_second_frame=np_Raw_Data[np.where(np.logical_and(np_Raw_Data[:,
0]>=58.0, np_Raw_Data[:, 0]<=70.0))]
    one_second_frame=np_Raw_Data[np.where(np.logical_and(np_Raw_Data[:,
0]>=currentSecond, np_Raw_Data[:, 0]<=currentSecond+1.0))]
    if one_second_frame.size>1:
        packetcount.clear()
        TotalPacketDuration.clear()
        TotalPacketLenght.clear()
        DioCount.clear()
        DisCount.clear()
        DaoCount.clear()
        src_duration.clear()
        dst_duration.clear()
        totalpackets=0
        frame_packet_length_sum=0
        total_duration=0.0
        src_packet_lenght_sum.clear()
        dst_packet_lenght_sum.clear()
        src_count.clear()
        dst_count.clear()
        for packet in one_second_frame:
            if not pd.isnull(packet[2]):
                src=packet[2]
                dst=packet[3]
```

```
            src_dst = src+"-"+dst
            packetcount[src_dst] = 1 if src_dst not in packetcount else packetcount[src_dst] + 1
            TotalPacketDuration[src_dst]=packet[1] if src_dst not in TotalPacketDuration else TotalPacketDuration[src_dst] + packet[1]
            TotalPacketLenght[src_dst]=packet[5] if src_dst not in TotalPacketLenght else TotalPacketLenght[src_dst] + packet[5]
            src_count[src]=1 if src not in src_count else src_count[src] + 1
            dst_count[dst]=1 if dst not in dst_count else dst_count[dst] + 1
            src_duration[src]=packet[1] if src not in src_duration else src_duration[src] + packet[1]
            dst_duration[dst]=packet[1] if dst not in dst_duration else dst_duration[dst] + packet[1]
            total_duration+=packet[1]
            src_packet_lenght_sum[src]=packet[5] if src not in src_packet_lenght_sum else src_packet_lenght_sum[src] + packet[5]
            dst_packet_lenght_sum[dst]=packet[5] if dst not in dst_packet_lenght_sum else dst_packet_lenght_sum[dst] + packet[5]
            frame_packet_length_sum+=packet[5]
            totalpackets+=1
            if packet[6]=="RPL Control (DODAG Information Object)":
               DioCount[src_dst]=1 if src_dst not in DioCount else DioCount[src_dst] + 1
            if packet[6]=="RPL Control (DODAG Information Solicitation)":
               DisCount[src_dst]=1 if src_dst not in DisCount else DisCount[src_dst] + 1
            if packet[6]=="RPL Control (Destination Advertisement Object)":
               DaoCount[src_dst]=1 if src_dst not in DaoCount else DaoCount[src_dst] + 1
            if ((packet[6]!="RPL Control (Destination Advertisement Object)") and (packet[6]!="RPL Control (DODAG Information Object)") and (packet[6]!="RPL Control (Destination Advertisement Object)")) :
               OtherMsg[src_dst]=1 if src_dst not in OtherMsg else OtherMsg[src_dst] + 1

      for i in packetcount:
         if not i in DioCount:
            arr_diocount=0
         else:
            arr_diocount=DioCount[i]
         if not i in DisCount:
            arr_discount=0
         else:
            arr_discount=DisCount[i]
         if not i in DaoCount:
            arr_daocount=0
         else:
            arr_daocount=DaoCount[i]
         if not i in OtherMsg:
            arr_orhermsg=0
```

```
            else:
               arr_orhermsg=OtherMsg[i]
            x=i.split("-")
            sourcee=x[0]
            destinatt=x[1]
            src_ratio=src_count[sourcee]/totalpackets
            dst_ratio=dst_count[destinatt]/totalpackets
            src_duration_ratio=src_duration[sourcee]/total_duration
            dst_duration_ratio=dst_duration[destinatt]/total_duration
            src_packet_ratio=src_packet_lenght_sum[sourcee]/frame_packet_length_sum

dst_packet_ratio=dst_packet_lenght_sum[destinatt]/frame_packet_length_sum

array=np.array([np.single(currentSecond),ip_dict[sourcee],ip_dict[destinatt],int(pack
etcount[i]),np.single(src_ratio),np.single(dst_ratio),np.single(src_duration_ratio),np.s
ingle(dst_duration_ratio),TotalPacketDuration[i],TotalPacketLenght[i],np.single(src_
packet_ratio),np.single(dst_packet_ratio),arr_diocount,arr_discount,arr_daocount,arr
_orhermsg,lbl],dtype="object")
            a_series = pd.Series(array, index = row.columns)
            row = row.append(a_series, ignore_index=True)
      currentSecond += 1.0
      counter+=1
      print(str(counter)+" of "+str(duration)+" of process is ok!!!")

row.to_csv(resultfile,index=False, sep=";")
```

**EK 4: Lojistik Regresyon, Rastgele Orman, Karar Ağaçları, Navie Bayes, KNN Sınıflandırıcı Algoritmaları İçin Yazılan Python Kodları**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time

def current_milli_time():
    return round(time.time() * 1000)

# Obtaining dataset with malicious mote
flnme1="DR-9N1M1R.csv"
# Obtaining dataset with normal motes
flnme2="DR-10N1R.csv"

df1=pd.read_csv("Results/New/"+flnme1, sep=";")
df2=pd.read_csv("Results/New/"+flnme2, sep=";")

# Obtaining row numbers of datasets
len1 = len(df1)
len2 = len(df2)

# This if block obtains equal rows from each dataset
if len1 < len2:
    df2 = df2.iloc[0:len1]
else:
    df1 = df1.iloc[0:len2]


# Merging two datasets
frames = [df1, df2]
df = pd.concat(frames)

X=df.iloc[:,3:16].values
y=df.iloc[:,16:17].values.ravel()

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.33,random_state=0)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

lrstart_time=current_milli_time()
from sklearn.linear_model import LogisticRegression
logr = LogisticRegression(random_state=0)
logr.fit(x_train,y_train)
```

```
lrend_time=current_milli_time()
LRduration=lrend_time-lrstart_time

y_pred_lr = logr.predict(x_test)


from sklearn.metrics import confusion_matrix
cm_lr=confusion_matrix(y_test,y_pred_lr)

rfstart_time=current_milli_time()
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_estimators=8, criterion='entropy')
rfc.fit(x_train,y_train)
rfend_time=current_milli_time()
RFduration=rfend_time-rfstart_time

y_pred_rfc = rfc.predict(x_test)

cm_rfc=confusion_matrix(y_test,y_pred_rfc)

dtstart_time=current_milli_time()
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier(criterion='entropy')
dtc.fit(x_train,y_train)
dtend_time=current_milli_time()
DTduration=dtend_time-dtstart_time

y_pred_dtc = dtc.predict(x_test)

cm_dtc=confusion_matrix(y_test,y_pred_dtc)


fptstart_time=current_milli_time()
from fylearn.fpt import FuzzyPatternTreeClassifier
fpc4=FuzzyPatternTreeClassifier()
fpc4.fit(x_train,y_train)
y_pred_fpc4 = fpc4.predict(x_test)

cm_fpc4=confusion_matrix(y_test,y_pred_fpc4)

fptend_time=current_milli_time()
FPTduration=fptend_time-fptstart_time

nbstart_time=current_milli_time()
from sklearn.naive_bayes import GaussianNB
gnb=GaussianNB()
gnb.fit(x_train,y_train)
y_pred_nb = gnb.predict(x_test)

cm_nb=confusion_matrix(y_test,y_pred_nb)
```

```
nbend_time=current_milli_time()
NBduration=nbend_time-nbstart_time


knnstart_time=current_milli_time()
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(x_train,y_train)
knnend_time=current_milli_time()
knnuration=knnend_time-knnstart_time

y_pred_knn = knn.predict(x_test)

cm_knn=confusion_matrix(y_test,y_pred_knn)
```

## EK 5: Yapay Sinir Ağı İçin Yazılan Python Kodları

```
import pandas as pd

# Obtaining dataset with malicious mote
flnme1="VN-9N1M1R.csv"
# Obtaining dataset with normal motes
flnme2="VN-10N1R.csv"

df1=pd.read_csv("Results/New/"+flnme1, sep=";")
df2=pd.read_csv("Results/New/"+flnme2, sep=";")

# Obtaining row numbers of datasets
len1 = len(df1)
len2 = len(df2)

# This if block obtains equal rows from each dataset
if len1 < len2:
   df2 = df2.iloc[0:len1]
else:
   df1 = df1.iloc[0:len2]


# Merging two datasets
frames = [df1, df2]
df = pd.concat(frames)

x=df.iloc[:,3:16].values
y=df.iloc[:,16:17].values.ravel()

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=0)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import time

def current_milli_time():
   return round(time.time() * 1000)


#classifier.add(Dense(3,kernel_initializer="glorot_uniform",
activation="relu",input_shape=(6,)))
```

```
model = keras.Sequential(
    [
        keras.Input(shape=(13)),
        layers.Dense(50, activation="relu"),
        layers.Dense(100, activation="relu"),
        layers.Dense(300, activation="relu"),
        layers.Dense(100, activation="relu"),
        layers.Dense(50, activation="relu"),
        layers.Dense(1, activation="sigmoid"),
    ]
)

start_time=current_milli_time()
model.compile(optimizer="Nadam",
loss="binary_crossentropy",metrics=['binary_accuracy'])

model.fit(x_train,y_train,epochs=60)
end_time=current_milli_time()
DLtime=end_time-start_time

y_pred_dl=model.predict(x_test)

y_pred_dl=(y_pred_dl>0.7)

from sklearn.metrics import confusion_matrix
cm_dl=confusion_matrix(y_test,y_pred_dl)
```

# ÖZGEÇMİŞ

**Kişisel Bilgiler**

| | |
|---|---|
| Soyadı, adı | : KİRAZ,Murat Uğur |
| Uyruğu | : T.C. |
| Doğum tarihi ve yeri | : |
| Telefon | : |
| e-mail | : |
| Web Sayfası | : |

**Eğitim**

| **Derece** | **Üniversite ve Bölüm** | **Mezuniyet tarihi** |
|---|---|---|
| **Lisans** | Kara Harp Okulu, Sistem Müh. | 2007 |

**Meslekî Deneyim**

| **Yıl** | **Yer** | **Görev** |
|---|---|---|
| 2008-2019 | Kara Kuvvetleri Komutanlığı | Tk./Bl.K.lığı |
| 2019-Şimdi | İtalya | Yazılım Uzmanı |

**Yabancı Dil**

İngilizce