
GAZİ ÜNİVERSİTESİ
Gazi
1926

KABLOSUZ SENSÖR AĞLARDA AĞ KATMANINDA MEYDANA GELEN DOS SALDIRILARININ DERİN ÖĞRENME YÖNTEMLERİYLE TESPİT EDİLMESİ

Celil OKUR

YÜKSEK LİSANS TEZİ
BİLGİ GÜVENLİĞİ MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım 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.

Celil OKUR
28/06/2022

KABLOSUZ SENSÖR AĞLARDA AĞ KATMANINDA MEYDANA GELEN DOS SALDIRILARININ DERİN ÖĞRENME YÖNTEMLERİYLE TESPİT EDİLMESİ

(Yüksek Lisans Tezi)

Celil OKUR

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2022

ÖZET

Kablosuz Sensör Ağlarının (KSA) kullanım alanları ve entegre olduğu sistem sayısı arttıkça popülaritesi artmakta ve bu durum saldırganların da dikkatini çekmektedir. KSA çalışma ortamı ve imkânları nedeniyle saldırıya açık bir yapıya sahiptir. Saldırganlar sensör ağlarına ağ içinden ve ağ dışından olmak üzere iki yolla; ağa sızmaya, ele geçirmeye, manipüle etmeye yönelik saldırılar gerçekleştirmektedirler. Gerçekleştirilen bu saldırılar katmanlara göre farklı şekilde uygulanmaktadır. Sensör ağlarda gerçekleştirilen bu saldırıların ardından ağ trafik verileri incelenmekte ve sonraki muhtemel saldırıların önüne geçmek için zararlı trafik ve düğüm davranışları analiz edilmektedir. Ağdan alınan ham veriler bazı ön işlemlerden geçirilerek öğrenme modelleri tarafından kullanılabilecek hale getirilmektedir. Modellerle incelenen veri, ağ trafik türlerine göre kategorize edilerek ağda gerçekleştirilen saldırılar tespit edilmektedir. KSA'da öğrenme modelleri ile yapılan saldırı tespitleri klasik tespit yöntemlerine göre yüksek doğruluk yüzdeleri ile gerçekleştirilmektedir. Bu tez çalışmasında KSA ağ katmanı saldırılarından olan Blackhole, Flooding, Selective Forwarding saldırıları Network Senario (NS 2) benzetim ortamında oluşturularak uygulanmış, WSN-BFSF veri seti elde edilmiş, elde edilen veri seti gerekli ön işlemlerden geçirilerek öğrenme modelleri ile incelemeye hazır hale getirilmiştir. Random Forest, Decision Tree, Naive Bayes ve Logistic Regression olmak üzere 4 farklı makine öğrenme modeli ve Multilayer Perception (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) ve Gated Recurrent Unit (GRU) olmak üzere 4 farklı derin öğrenme modeli ile toplamda 8 farklı modelle veri seti incelenmiştir. Modeller ile elde edilen deneysel sonuçlar detaylı olarak sunulmuştur.

Bilim Kodu	:	92403
Anahtar Kelimeler	:	Kablosuz Sensör Ağları, DoS-DDoS, Saldırı Tespiti, Güvenlik, Derin Öğrenme
Sayfa Adedi	:	86
Danışman	:	Doç. Dr. Murat DENER

DETECTION OF DOS ATTACKS OCCURING AT NETWORK LAYER WITH DEEP LEARNING METHODS IN WIRELESS SENSOR NETWORKS

(M. Sc. Thesis)

Celil OKUR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2022

ABSTRACT

As the usage areas of wireless sensor networks and the number of systems it is integrated increase, their popularity also increases which attracts the attention of attackers. WSN has a structure open to the attacks due to its working environment and opportunities. Attackers carry out attacks to infiltrate, seize, and manipulate sensor networks both inside and outside the network. These attacks are carried out differently based on the layers. After these attacks on sensor networks, network traffic data is examined and malicious traffic and node behaviors are analyzed to prevent possible future attacks. The raw data received from the network is made usable by learning models by passing some preprocessing. The data analyzed with the models are categorized according to the network traffic types and the attacks carried out on the network are detected. In WSN, attack detections made with learning models are performed with high accuracy percentages compared to classical detection methods. In this study, the data set was obtained by performing Blackhole, Flooding, Selective Forwarding attacks, which are network layer attacks in WSN in the Network Senario (NS 2) simulation environment, and the obtained data set was made ready for examination with learning models after the necessary preprocessing. The data set was examined with four different machine learning models, namely Random Forest, Decision Tree, Naive Bayes and Logistic Regression, and four different deep learning models, namely Multilayer Perception (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) with 8 different models. Result of models are explained in detail.

Science Code	:	92403
Key Words	:	Wireless Sensor Network, DoS-DDoS, Attack Detection, Security, Deep Learning
Page Number	:	86
Supervisor	:	Assoc. Prof. Dr. Murat DENER

TEŞEKKÜR

Öncelikle tanıdığım günden beri üzerimde büyük emeği olan ve tez yazım süreci boyunca bana göstermiş olduğu sabır, teşvik, rehberlik ve geri bildirimlerinden dolayı değerli hocam, danışmanım Doç. Dr. Murat DENER'e teşekkürlerimi sunarım. Bu tezin yazım süreci boyunca tıpkı zor zamanlarımda olduğu gibi bu zorlu süreçte de tüm anlayış ve sabrı ile yanımda olan değerli eşim Seda OKUR, hayatımıza varlığıyla renk katan biricik kızım Latte'ye ve bugünlere getiren değerli aileme teşekkür ederim. En önemlisi bu tezi HER ŞEY'e rağmen bitirebilmiş olduğum için en büyük teşekkürü kendime etmek istiyorum. Bu tez kendime ve yeni hayatıma armağan olsun. Gelecek güzel günler için umudumu kaybetmeden çalışmaya, üretmeye devam edeceğim; daha güzel daha üretken günlere.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. KABLOSUZ SENSÖR AĞLARDA GÜVENLİK 5

2.1. Kablosuz Sensör Ağlar 5

2.2. Kablosuz Sensör Ağlarının Karakteristik Özellikleri 7

2.2.1. Geniş ölçeklendirme 7

2.2.2. Sınırlı kaynak 7

2.2.3. Fazlalık 7

2.2.4. Güvenlik 7

2.2.5. İletişim kabiliyeti 8

2.2.6. Çok adımlı iletişim 8

2.2.7. Dinamik 8

2.2.8. Oto organizasyon 9

2.2.9. Uygulama uyumluluğu 9

2.2.10. Düşük maliyet 9

2.3. Kablosuz Sensör Ağlarında Güvenlik Gereksinimleri 9

2.3.1. Veri gizliliği 10

Sayfa

2.3.2. Veri bütünlüğü 10
2.3.3. Erişilebilirlik 10
2.3.4. Veri tazeliği 10
2.3.5. Kaynak doğrulama 11
2.3.6. Kullanılabilirlik 11
2.3.7. İnkar edilemezlik 11
2.3.8. Erişim kontrolü 12
2.3.9. Güvenli konumlanma 12
2.3.10. Zaman senkronizasyonu 12
2.3.11. Servis kalitesi 13
2.3.12. İleri-geri gizlilik 13
2.4. Kablosuz Sensör Ağlarda Güvenlik Zafiyetleri ve Saldırı Türleri 13
3. LİTERATÜR ÇALIŞMALARI 17
4. ATAK MODELLERİ 23
4.1. Blackhole Saldırısı 23
4.2. Flooding Saldırısı 26
4.3. Selective Forwarding Saldırısı 29
5. YAPAY ZEKA VE TÜRLERİ 33
5.1. Yapay Zeka 33
5.2. Makine Öğrenmesi 33
5.2.1. Decision Tree 34
5.2.2. Random Forest 35
5.2.3. Naive Bayes 36
5.2.4. Logistic Regression 36

Sayfa

5.3. Derin Öğrenme 36

5.3.1. Derin öğrenme modelleri 38

5.3.2. Derin öğrenme fonksiyonları 41

6. ÇALIŞMADA KULLANILAN ARAÇLAR 45

6.1. NS-2 45

6.2. Ağ Senaryo Üreticisi 46

6.3. Google Collabratory 47

6.4. Apache Spark 48

7. GELİŞTİRİLEN UYGULAMA 51

8. WSN-BFSF VERİ SETİ ÖZELLİKLERİ 61

9. MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ANALİZLERİ 63

9.1. Veri Ön İşleme 63

9.2. Analiz İşlemleri 65

10. SONUÇ VE ÖNERİLER 75

KAYNAKLAR 76

EKLER 82

EK-1. Veri Seti Elde Etme Aşamaları 83

EK-2. Veri Dağılımı 84

ÖZGEÇMİŞ 85

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. Literatür özeti 13

Çizelge 3.2. KSA alanında kullanılan veri setleri 21

Çizelge 7.1. Senaryo parametreleri 55

Çizelge 8.1. WSN-BFSF veri setinin özellik açıklaması 61

Çizelge 9.1. WSN-BFSF veri dağılımı 63

Çizelge 9.2. Sayısal dönüşüm işlemi 64

Çizelge 9.3. GRU algoritması ile kullanılan hiperparametreler 69

Çizelge 9.4. Accuracy, Precision, Recall ve F-Score sonuçları 70

Çizelge 9.5. GRU algoritmasının sınıflandırma performansı 70

Çizelge 9.6. Algoritmaların trafik sınıflarına göre doğruluk değer performansı 73

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 4.1. Blackhole saldırı modeli 24

Şekil 4.2. Blackhole saldırısının psuedo kodu 25

Şekil 4.3. Flooding saldırı modeli 27

Şekil 4.4. Flooding saldırısının psuedo kodu 28

Şekil 4.5. Selective Forwarding saldırı modeli 30

Şekil 4.6. Selective Forwarding saldırısının psuedo kodu 30

Şekil 5.1. Decision Tree algoritma şeması 34

Şekil 5.2. Random Forest algoritma şeması 35

Şekil 5.3. Convolutional Neural Network algoritma şeması 38

Şekil 5.4. Multilayer Perceptron algoritma şeması 39

Şekil 5.5. LSTM algoritma şeması 40

Şekil 5.6. GRU algoritma şeması 41

Şekil 6.1. NS-2 benzetim ortamı arayüzü 45

Şekil 6.2. NSG aracı arayüzü 46

Şekil 6.3. Google Collabratory 48

Şekil 6.4. Apache Spark kullanım kodları 50

Şekil 6.5. Apache Spark mimarisi 50

Şekil 7.1. AODV çalışma mimarisi 51

Şekil 7.2. NS-2 arayüzü 53

Şekil 7.3. Network Senario Generator arayüzü 54

Şekil 7.4. Düğümlerin durumlara göre enerji kullanım miktarı 55

Şekil 7.5. Senaryo akış şeması 57

Şekil **Sayfa**

Şekil 8.1. WSN-BFSF veri setinin elde etme aşamaları 62

Şekil 9.1. WSN-BFSF veri için Pearson Correlation Matrix sonuçları 66

Şekil 9.2. Makine/derin öğrenme akış şeması 67

Şekil 9.3. Modellerin doğruluk yüzde sonuçları 68

Şekil 9.4. Precison, Recall ve F1-Score parametrelerine göre karşılaştırma sonuçları .. 70

Şekil 9.5. Confusion matrix 71

Şekil 9.6. ROC eğrisi 72

Şekil 9.7. Precision-Recall eğrisi 72

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ADC	Analog to Digital Converter
ANN	Artificial Neural Network
AÜD	Ağ Üye Düğümü
AODV	Adhoc On-Demand Distance Vector
BD	Bilinmeyen Düğüm
CNN	Convolutional Neural Network
CSMA	Carrier Sense Multiple Access
DoS	Denial of Service
DDoS	Distributed Denial of Service
DN	Düğüm Numarası
DSR	Dynamic Source Routing
DZ	Zaman Değişimi
ELU	Exponential Linear Unit
FNR	False Negative Rate
FN	False Negative
FP	False Pozitive
FPR	False Pozitif Rate
G	Ortalama RREQ Zamanı
GRU	Gated Recurrent Unit
HTC	Human Type Communication
IDS	Intrusion Detection System
IF	Isolation Forest
IoT	Internet of Things
IPS	Intrusion Prevetion System
KNN	K-Nearest Neighbors

Kısaltmalar	Açıklamalar
KSA	Kablosuz Sensör Ağları
LOF	Local Outlier Factor
LR	Logistic Regression
LSTM	Long Short Term Memory
MAC	Message Authentication Code
MAC	Medium Access Control
MANET	Mobile Adhoc Network
MAÜD	Muhtemel Ağ Üye Düğümü
MITM	Man In The Middle
ML	Machine Learning
MLP	Multilayer Perceptron
MTC	Machine Type Communication
N-BaIoT	Network-Based Detection of IoT Botnet Attacks
NSG	Network Senario Generator
NSL-KDD	Network Security Laboratory Knowledge
NS-2	Network Senario 2
OMNET	Objective Modular Network
PCA	Principal Component Analysis
QoS	Quality of Service
R	Düşürülen Paketi Hesaplayan Fonksiyon
R2L	Remote To Local
ReLu	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RREQ	Route Request
SD	Sensör Düğüm
SHIoT	Smart Home Internet of Things
SVM	Support Vector Machine
SWSNM	Secure Wireless Sensor Network Middleware
TCL	Tool Command Language
TP	True Positive

Kısaltmalar	Açıklamalar
TN	True Negative
U2L	User To Local
WEKA	Waikato Environment for Knowledge Analysis
WSN	Wireless Sensor Network
WSN-BFSF	Wireless Sensor Network Blackhole Flooding Forwarding
WSN-DS	Wireless Sensor Network Data Set
X	Düğüm Numarası
Z	Zaman
ZD	Zararlı Düğüm

1. GİRİŞ

Kablosuz Sensör Ağlarının (KSA) son yıllardaki gelişmeler ile birlikte kullanım alanları ve sektördeki önemi artmıştır. Amaca uygun olarak birçok alanda kullanılabilen, modüler, kullanıcı dostu, kullanıcıya pratik çözümler sunan ve alt yapı gerektirmeyen bir sistemdir [1-3]. KSA market hacminin 2024 de 1.8 milyon dolara ulaşması beklenmektedir [4, 5]. KSA kısaca kablosuz servis ile çevrenin algılanmasını sağlayan sensörlerden oluşan senkronize çalışan bir organizasyondur [1, 6]. Bulunduğu ortamın şartlarına göre değiştirilip ortama entegre edilebilmektedir. KSA sağladığı bu kolaylık açısından bilgi sistemlerinden ayrılmaktadır. Bilgi sitemleri her ne kadar kullanıcıya günlük hayatta kolaylıklar sağlasa da kurulumu, sürdürülebilirliği ve bakımı gibi durumlardan dolayı KSA ya göre zahmetlidir. Her ne kadar KSA da merkezi bir sistem ile çalışsa da bilgi sistemleri kadar mutlak bir merkezi yönetim ile idare edilmemektedir.

Özellikle oto senkron olması her düğüm için ayrı kurulum gerektirmemektedir. Bu özelliği ile Internet of Things (IoT) gibi birçok teknoloji ve sistemle birlikte kullanılmaktadır. Özellikle günümüzde her alanda kullanılan sensör ağ teknolojisi akıllı evler, akıllı şehirler gibi projelerin yanı sıra IoT teknolojisi ile entegre kullanıldığında her seviyede kullanıcının ilgisini çekmektedir. Askeri alan, doğal yaşam, endüstri, sağlık ve gündelik hayat gibi birçok farklı alanda uygulamaları görülmektedir [7]. Bu kadar geniş ve farklı alanda kullanılan bir sistemin güvenlik zafiyetleri saldırganlar tarafından sorgulanmaktadır. Her ne kadar güvenliği sağlansa da sensör ağlar kullanıldıkları ortam ve protokollere bağlı olarak bazı güvenlik problemlerini barındırmaktadır.

KSA özellikle askeri ve kritik bölgelerde kullanıldığında güvenlik daha büyük bir önem arz etmektedir. Sensörlerin çalışma ortamları bazen dış mekân olduğunda saldırılara karşı daha korunmasız hale gelmektedir. Fiziki saldırıları bazı tedbirlerle engellemek mümkün olmaktadır. Diğer taraftan saldırılar bazen iç bazen de dış ağdan gelebilmektedir. Bu saldırıları belirlenen ölçütlere göre gruplamak mümkündür. Yapılan bu saldırılar sistem ve kullanıcılara birtakım zararlar vermektedir. Her ne kadar çalışma prensibi bilgisayar ağlarıyla benzerlik gösterse de saldırılardan bilgisayar ağlarına göre daha çok etkilenmektedir. Bilginin gizliliği, bütünlüğü ve erişilebilirliği bilgi güvenliliğinin en temel unsurlarıdır. Bu unsurlar KSA için de geçerlidir. Dolayısıyla güvenli bir KSA da bu

gereklerin yerine getirilmesi beklenmektedir. Sensör ağlarının özellikle dış ve ulaşılması zor ortamlarda altyapı, periyodik bakım ve güncelleme gibi gereklerinin olmaması kullanıcıya büyük kolaylık sağlamaktadır. Ancak sensör ağları bakımından bu durum sınırlı kaynak problemini beraberinde getirmektedir. Bellek, işlemci, enerji gibi özelliklerin en ideal şekilde kullanılması KSA için büyük önem arz etmektedir. Bunun için ihtiyaca yönelik doğru senaryoların uygulanması gerekmektedir. Buna ek olarak bu sınırlı kaynak problemi saldırganların da dikkatini çekmektedir. Bu problemden yararlanarak geliştirilen çeşitli saldırılar, ağın kullanım ömrünü kısaltmak, ağı kullanılamaz hale getirmek, iletişimi manipüle etmek, veri çalmak gibi sorunlara neden olmaktadır. Saldırılar fiziksel ve yazılımsal olarak gerçekleştirilmektedir. Bilgisayar ağlarından farklı olarak sensör ağlar hem iç hem dış ortamlarda da çalışmaktadır. Fiziksel olarak ulaşılabilir olması kötü niyetli erişimlerde ağa ve iletişime zarar vermektedir. Çalışma şartlarına göre çoğu zaman fiziksel olarak korumasız ortamlarda çalıştırılmaktadır. Bu durumda fiziksel saldırıya imkan sağlamaktadır. Yazılımsal olarak gerçekleştirilen saldırılar içerden ve dışardan gerçekleştirilmektedir. Dışardan gerçekleştirilen saldırılar kapalı ağ sistemine dışardan yeni bir düğüm ile ağa dahil olunarak yapılan saldırılardır. Genellikle yazılımsal saldırılar ağ üzerinden gerçekleştirildiği için ağ saldırıları olarak adlandırılmaktadır. İçerden gerçekleştirilen saldırılar; mevcut çalışan düğümün çeşitli yollar ile ele geçirilerek saldırgan tarafından kullanılması ile gerçekleştirilmektedir. Saldırılar amaçlarına göre ve gerçekleştirildiği katmanlara göre farklılık göstermektedir. Bu çalışmada ağ katmanı saldırılarından olan Blackhole, Flooding ve Selective Forwarding atakları NS-2 benzetim ortamında gerçekleştirilmiştir. Saldırıların gerçekleştirilmesinin ardından trafiğe ait tüm veriler alınarak WSN-BFSF veri setinin ham hali kaydedilmiştir. WSN-BFSF veri seti saldırı trafik verilerinden (Blackhole, Flooding, Selective Forwarding) ve Normal trafik verilerinden oluşmaktadır. Kaydedilen ham veri seti makine ve derin öğrenme modelleri ile doğrudan kullanılmamaktadır. Bu yüzden aşamalı olarak ön işlemlerden geçirilerek modellerde kullanılabilir hale getirilmişlerdir. Öğrenme modelleri ile analiz edilen veri, ağ trafik türlerine göre sınıflandırılarak sonuçlar kaydedilmiştir. Kısacası çalışmada hem veri seti üretilmiş hem de veri seti öğrenme modelleri ile kullanılarak yüksek doğruluk yüzdeleri ile saldırı tespiti yapılmıştır.

Gerçekleştirilen çalışmanın literatüre katkısı:

- KSA benzetim ortamında saldırıların nasıl yapılacağının aşama aşama anlatılmış olması,
- Oluşturulan WSN-BFSF veri seti ile KSA alanında sınırlı sayıda olan veri seti sayısının

arttırılmış olması,

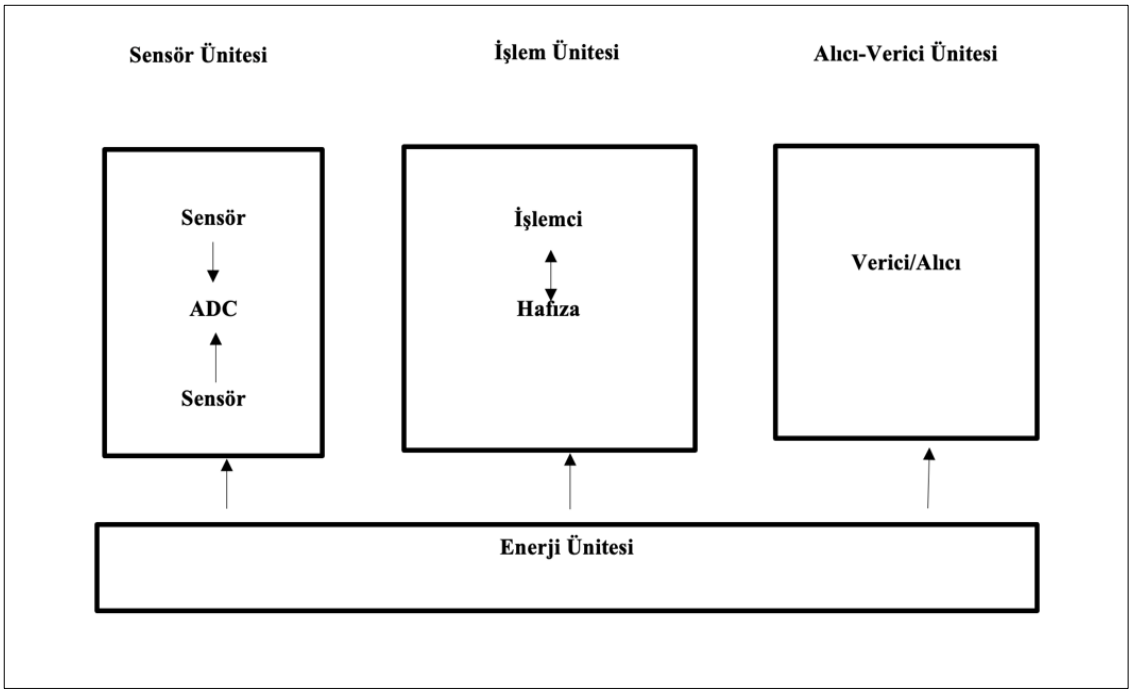
- Yeni veri setleri oluşturulurken yol gösterici olması bakımından veri setinin oluşturulma aşamalarının detaylı bir şekilde anlatılması,
- Çalışmada WSN-BFSF veri setinin hem makine öğrenme modelleri ile hem de derin öğrenme modelleri ile yüksek doğruluk sonuçları elde edilerek incelenmesi şeklinde sıralamak mümkündür.

Çalışmanın ikinci bölümünde KSA genel özellikleri, karakteristiği, yapılan saldırılar ve güvenlik gereksinimlerinden bahsedilmektedir. Üçüncü bölümde ise çalışmada gerçekleştirilen saldırı modellerinden detaylı olarak bahsedilmiştir. Dördüncü bölümde Literatürde taraması yapılarak yapılan ilgili çalışmalar incelenmiştir. Beşinci bölümde ise yapay zeka ve alt dalları olan makine öğrenmesi ve derin öğrenme modelleri, fonksiyonları incelenmiştir. Altıncı bölümde çalışmada kullanılan araçlardan detaylı olarak bahsedilmiştir. Yedinci bölümde uygulamada veri setinin elde edilmesi ve aşamaları anlatılmıştır. Sekizinci bölümde veri setinin özelliklerinden bahsedilmiştir. Dokuzuncu bölümde veri ön işlemesinden, öğrenme modelleri ile verinin analiz edilmesinden ve sonuçlarından bahsedilmiştir. Onuncu bölümde çalışmanın sonuçlarına ve önerilere yer verilmiştir.

2. KABLOSUZ SENSÖR AĞLARDA GÜVENLİK

2.1. Kablosuz Sensör Ağları (KSA)

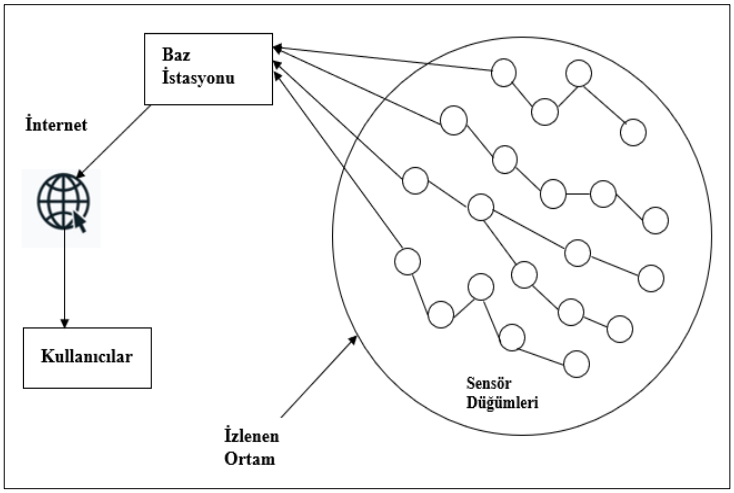
Kablosuz sensör ağları kablosuz servis ile bulunduğu ortamdaki algılanabilir değişmeleri sensör kabiliyetleri dâhilinde algılayıp, alt yapı gerektirmeden sınırlı kaynaklarını kullanarak kullanıcıya sunan sistemlerdir [6, 8]. Sensörler kabiliyetleri dahilinde ortamlardaki değişmeleri algılayarak, önceden belirlenen iletişim mimarisine göre algıladığı verileri baz istasyonuna doğrudan ya da aktarmalı olarak göndermektedirler. Baz istasyonu, kullanıcı ile sensör ağı arasındaki iletişimin sağlandığı köprüdür. Sensör ağları ortamın gerekleri ve ihtiyacına göre farklı sayıda kullanılabilir. Bu sistemin ölçeklenebilir olduğunu göstermektedir. İnsanların ulaşamadığı ya da güvenlik bakımından ulaşılabilir olmayan yerlerden veri almak istediğinde sensör ağlarının önemi daha net görülmektedir. Şekil 2.1. sensör düğüme ait şema gösterilmiştir.



Şekil 2.1. Sensör düğüm şeması

Şekil 2.1'de görüldüğü gibi sensörler farklı ünitelerden oluşmaktadır. Ortamdaki değişimler sensörün kabiliyetine göre algılanarak Analog to Digital Convertor (ADC) ünitesine iletilmektedir. ADC ünitesinde, gelen verilere sayısal dönüşüm işlemi uygulanmaktadır.

ADC ünitesinden sayısal veriler işlemci bölümüne iletilmektedir. İşlemci bölümünde gerekli aşamalardan geçirilerek gönderime hazır hale getirilmektedir. İşlemci bölümünden sayısal veriler gönderilmek üzere alıcı-verici ünitesine iletilmektedir. KSA'nın ekstra alt yapı, araç ve kaynak gerektirmemesi farklı alanlarda kullanımını arttırmıştır. Teknik özelliklerine göre iç ve dış ortamlarda (hava, kara, sualtı, yeraltı) kullanılabilmektedir. Dış ortamlarda kullanılabilir ve sürdürülebilir olması KSA'nın askeri, doğal yaşam ve endüstriyel alanlarda çok tercih edilen bir teknoloji olmasını sağlamıştır. KSA mimari yapısı Şekil 2.2'de gösterilmiştir. Bu avantajlarının yanında enerji, işlemci, hafıza gibi konularda kapasitesinin sınırlı olması sensör ağlarının dezavantajları olarak söylenebilmektedir [8]. Bir ortamda bulunan farklı ve çok fazla sayıdaki sensör düğümleri ortamdaki bir değişimi aynı şekilde algılayıp veri üretebilmektedirler. Bu özellik düğümlerin doğrulanabilirliği açısından avantaj gibi görünse de ağ için yoğun bir trafiğe yol açmaktadır. Aynı veriden çok fazla olması tekrar paket gönderimleri ağın kaynaklarının çabuk tüketilmesine ve ağda sıkışmalara yol açmaktadır. Bu durum sensör ağları için dezavantaj olarak görülmektedir [7, 9]. Bunların yanı sıra KSA'nın fiziksel ve yazılımsal güvenliği konusunda da kısıtları bulunmaktadır. Klasik bilgisayar ağlarında olduğu gibi bütün güvenlik yazılımları kaynak kısıtlaması nedeniyle istenildiği gibi kullanılamamaktadır. Sensör ağlar için güvenlik modeli geliştirirken her zaman kaynakların sınırlı olduğu düşünülmekte ve ona göre model optimize edilmektedir. Her ne kadar kullanımda güvenlik adımları fazladan yük gibi görülse de sensör ağlarda güvenlik için harcanan kaynaklar güvenlik zafiyetlerinden dolayı kaybedilen kaynaklardan çok daha az olmaktadır.



Şekil 2.2. Kablosuz sensör ağları mimarisi

Şekil 2.2'de görüldüğü gibi düğümler tarafından algılanan veriler ağ mimarisine göre ya küme başı düğüme ya da baz istasyonuna iletilmektedir. Baz istasyonunda toplanan veriler internet üzerinden kullanıcıya iletilmektedir.

2.2. KSA'nın Karakteristik Özellikleri

2.2.1. Geniş ölçeklendirme

KSA kullanıldığı ortamı algılanabilirlik açısından tamamen kontrol altına alması için, aynı ortamda seyrek- sık şekilde planlanabildiği gibi farklı boyutlardaki ortamlarda da farklı miktarlarda kullanılabilmektedir. Bu kullanıcıya miktar açısından modüler kolaylık sağlamaktadır. Sensör ağlarda düğüm sayısı 100.000'lerce düğüme kadar çıkarılabilmektedir [10].

2.2.2. Sınırlı kaynak

Sensör ağlar dış ortamda kullanıldığında önemi daha net anlaşılmaktadır. Özellikle ulaşılması güç ortamlarda sensör ağlara periyodik destek sağlanamadığından mevcut kaynakların minimum seviyede kullanılması ağın ömrünü doğrudan etkilemektedir. Aynı zamanda işlemci ve hafıza bakımından sınırlı olması kompleks işlemlerin yapılmasını engellemektedir. Sonuç olarak enerji, hafıza, işlemci, radyo iletişimi gibi kaynakların kısıtlı olması KSA'yı sınırlamaktadır [11].

2.2.3. Fazlalık

Ortamdaki farklı sensörler algılanabilir aynı değişiklikleri fark edip veri üretmekte ve kaynakları gereksiz yere tüketmektedir. Kısıtlı olan kaynaklar aynı amaç için gereksiz yere tüketilmektedir [12].

2.2.4. Güvenlik

Güvenlik her ne kadar önemli olsa da kaynakların kısıtlı olduğu sensör ağlarında kaynak tüketimi açısından bazen tercih edilmemektedir. Veri hassasiyetinin önemli olduğu ortamlarda (askeri, sağlık, farklı güvenlik endüstrisinde) güvenlik göz ardı edilmemektedir.

Ancak güvenliğin ikinci planda tutulduğu ortamlarda ise kaynakların sömürülmesine yönelik saldırılarda göz ardı edilen güvenlik, önem kazanmaktadır. Sensör ağlarda güvenliğin sağlanması ile sınırlı kaynakların tüketimi arasında bir paradoks bulunmaktadır. Geliştirici, kaynakları verimli kullanmak amacı ile güvenliği göz ardı ettiğinde her zaman kaynakların ve verilerin ilerleyen zamanlarda manipüle edilme ihtimali bulunmaktadır. Güvenliği sağladığında ise bazı ek önlemler (kaynak doğrulama, şifreleme, dijital imza) kaynak tüketimini arttırmaktadır. Bu ek önlemlerin sistemde kullanılıp kullanılmaması geliştiricinin tercihine bağlıdır [13].

2.2.5. İletişim kabiliyeti

Sensör ağlarda iletişim bant genişliği dar ve değişkendir. Düğümler arası uzaklık birkaç yüz metre ile sınırlıdır. Ayrıca sensör ağları yağmur, fırtına gibi meteorolojik değişikliklerden ve bina, dağ gibi dış etkenlerden kolayca etkilenebilmektedir. Bu yüzden sensör ağlarının sorunsuz olarak çalışabilirliğinin devamlı olması pek mümkün değildir. Bu yüzden sensör ağlarının donanımsal ve yazılımsal olarak dirençli, hata toleransı düşük ve güvenlik mekanizması yüksek olmalıdır [14].

2.2.6. Çok adımlı iletişim

Sensör ağları oluşturulurken fazla sayıda düğüm kullanılabilmektedir. Düğümler ortamların büyüklüklerine göre konumlandırıldıktan sonra farklı noktalarda bulunan düğümlerle iletişimi aralarında bulunan düğümler üzerinden gerçekleştirmektedirler. İletişim aracı ara düğümler üzerinden gerçekleştirilebilmektedir [15].

2.2.7. Dinamik

Limitli kaynaklar nedeniyle ağlarda bazı düğümler bazen ağ dışında kalır. Özellikle enerji bakımından düğümlerin ömürlerinin sonlanması durumunda ağa yeni alternatif düğümler eklenebildiği gibi kalan düğümler ile iletişim devam edebilmektedir. İletişimin devam etmesi için önceki düğümler üzerinden gerçekleşen iletişim artık yeni alternatif düğümler üzerinden gerçekleşmektedir. Topolojideki bu değişim KSA'nın dinamik ve oto senkronizasyon özelliği sayesinde iletişimin alternatif yollardan otomatik olarak gerçekleşmesini sağlamaktadır [16].

2.2.8. Oto-organizasyon

KSA teknolojisi kendi kendine senkron olarak çalışabilen alt yapı gerektirmeyen modüler yapılardır. Bu yönleri sensör ağlara önemli avantajlar sağlamakta ve kullanıcılar tarafından tercih edilmektedir. Sensör ağları kurulum aşamasında ağ topolojileri farklı olsa da belirli bir düzen ile hangi düğümün nerde çalıştığı, hangi düğümler üzerinden iletişimin gerçekleşebileceği, ağ düğümleri tarafından kendi aralarında haberleşmeleri sayesinde bilinmektedir. Bunun için geliştiricinin müdahalesi olmasına gerek yoktur. Oto organizasyon özelliği sayesinde ağa sonradan dahil olan düğümler dahi diğer eski üye düğümler gibi otomatik olarak çalışabilmektedir [17].

2.2.9. Uygulama uyumluluğu

KSA'da çok adımlı iletişim ara düğümler üzerinden sürdürülmektedir. Düğümler özelliklerine göre algıladıkları veriyi tek merkezde toplama mantığıyla iletimini sağlamaktadırlar. Veri iletim şekli çok düğümden tek düğüme olacak şekilde gerçekleşmektedir. Her sensör ağının kullandığı farklı iletişim uygulaması ve farklı uygulamanın da farklı işlem süreci olmaktadır. Yani uygulama temelli çalışan farklı sensör ağlarında bir ağda kullanılan yönlendirme uygulaması diğerinde etkili olmayabilmektedir [18].

2.2.10. Düşük maliyet

Takip edilmek için belirlenen ortam yüzlerce düğüm ile takip edilebildiği gibi binlerce düğüm ile de takip edilebilmektedir. Her düğümün maliyet arttırdığı düşünüldüğünde toplam maliyeti azaltmak için mümkün olan en az düğüm ile ortam takip edilip toplam maliyet düşürülebilmektedir [19].

2.3. KSA'da Güvenlik Gereksinimleri

Kablosuz Sensör Ağları'nın güvenlik gereksinimlerini aşağıda bulunan başlıklar altında sıralamak mümkündür [7].

2.3.1. Veri gizliliği

Verinin yetkisiz ya da ulaşmaması gereken kişiler tarafından ulaşılmasının engellenmesidir. Veri gizliliğinin sağlanması için veriler transfer edilirken ve saklanırken şifrelenmektedir. Bu şifrelemenin hem kırılabilmesinin zor olması gerekmekte hem de şifreleme için çok fazla enerjinin harcanmaması istenmektedir. KSA da verinin gizliliği veriyi birleştiren düğümler tarafından sağlanmaktadır. Veriyi gönderen sensör düğüm veriyi şifreleyerek veri birleştirici düğüme gönderir. Veriyi alan üst düğüm veriyi eski haline getirir. Birleştirme işleminin ardından tekrar birleştirilmiş veriyi şifreleyerek bir üst birleştiriciye ya da baz istasyonuna gönderir [20].

2.3.2. Veri bütünlüğü

Düğümler arası iletilen verinin değiştirilmesini ya da anlam bütünlüğünün bozularak anlamsız hale getirilmesini önlemektir. Sensör ağlarda içerde araya giren zararlı düğümler bu bozma ya da değiştirme işlemini gerçekleştirebilirler. Bunun önüne Message Authentication Code (MAC) ile geçilebilir [21].

2.3.3. Erişilebilirlik

Sistemin ve verinin zarar görmemesi ve verinin çalınmaması kadar sisteme erişilebilirliğin sağlanması da önemlidir. Bunun sağlanamaması için literatürde en çok karşılaşılan saldırı türü DoS-DDoS atak türleridir. Fiziksel ve ağ katmanında yapılabilen bir saldırı türüdür. Amaç sistemi kullanıcılara erişilmez kılmaktır. Sensör ağlarda buna ek olarak ağın enerji kaynaklarını tüketerek ağın faaliyetinin sonlandırılması hedeflenmektedir [18].

2.3.4. Veri tazeliği

Bu alanda verinin bütünlüğü ve gizliliği kadar verinin tazeliği de önemlidir. Sensörler ortamda kullanıcıların istediği nitelikleri ve değişimleri ölçerler. Yapılan ölçümlerin hangi zamana ait olduğu önemlidir. Özellikle saldırganlar eski ölçüm değerlerini güncelmiş gibi sunarak denetleme mekanizmasını manipüle edebilmektedir [22]. Ortamdaki sıcaklık ve oksijen değerlerini ölçen bir sistemin yangını haber verememesi ya da yanlış alarm vererek kullanıcıyı yanıltması veri tazeliğinin önemini açıklayan bir örnektir [23].

2.3.5. Kaynak doğrulama

Sensör ağların en önemli özelliği kendi kendine konfigüre olabilmeleridir. Bu özellik çoğunlukla bu alanın en önemli kolaylığı olsa da saldırganlar tarafından sıkça kullanılmaktadır. Oto konfigürasyon ile ağa sonradan katılan bir düğüm iletişime kolayca dahil olabilir [24]. Bu sayede saldırgan kendi düğümünü kolayca ağa dahil edebilir. Ayrıca bir düğüm kendini birden fazla düğüm gibi göstererek ağ trafiğini karıştırabilir. Bu yüzden her düğümün kimlik doğrulamasının yapılması önem arz etmektedir [25].

2.3.6. Kullanılabilirlik

Kullanılabilirlik KSA'da enerji ve güvenliğin temel sonucudur. Kullanılabilirlik ile kaynak yönetimi arasında doğrudan bir ilişki vardır. KSA kaynaklarının kullanım miktarı kaynak yönetiminin planlanmasına bağlıdır. Kısıtlı kaynakların düzgün ve güvenli bir planlaması sonucunda kullanılabilirlik sağlanmış olur. Kullanılabilirlik KSA da sınırlı kaynakların planlı yönetilmesi ile ilgilidir. Diğer taraftan sınırlı kaynakların zararlı erişimler tarafından aşırı tüketimini engellemek de kullanılabilirliği arttırmaktadır. Ağlara ait kaynakların manipüle edilerek tüketilmemesi ve adil dağıtılmaması kesintilere neden olmaktadır. Özellikle DoS saldırıları ağın kaynaklarını tüketmeye yönelik saldırılardandır. Saldırganların izinsiz ağa dahil olmalarını engellemek için yetkilendirme kontrolü gibi yöntemler kullanılarak kullanılabilirlik sağlanmaktadır. Kullanılabilirlik ağın devamlılığı açısından önem arz etmektedir [26].

2.3.7. İnkar edilemezlik

İnkar edilemezlik zararlı veya yetkisiz erişimlerin araya girerek verileri alıp değiştirme ya da eksik gönderme gibi saldırıları tespit etmesi açısından gereklidir. Veriyi gönderenin dijital imzası ile veri bütünlüğü korunurken aynı zamanda verinin kim tarafından gönderildiği de belirlenmiş olmaktadır [27]. Bilgisayar ağlarında olduğu gibi gönderici ve alıcı arasındaki iletişim dijital imza üzerinden gerçekleştirilmektedir. Bu durum gönderenin doğrulanmasının ve gönderim reddinin önüne geçilmesi açısından gereklidir [28].

2.3.8. Erişim kontrolü

Sensör ağlarının kullanımı ortamı bakımından fiziksel ve yazılımsal olarak saldırgan erişimine açıktır. Özellikle yetkisiz erişimler; ağ performansında azalmalara, verilerin gizliliğinin ihlal edilmesine, verilerin ve ağın manipüle edilmesine neden olmaktadır. Bu yüzden erişimlerin yetkilendirilmesi ve yetkisiz erişimlerin önüne geçilmesi gerekmektedir. Erişim kontrolü bu yüzden KSA güvenliği için gerekliliktir [18].

2.3.9. Güvenli konumlanma

KSA çalışma olarak iç ortamlarda tercih edildiği gibi dış ortamlarda da tercih edilmektedir. Her iki ortamda da kullanılırken saldırganlar tarafından ulaşılabilir konumda olabilmektedir. Özellikle baz istasyonu tüm ağa ve düğümlere ait verileri bulundurduğundan; bu düğümün ulaşılarak manipüle edilmesi, tüm ağ için problemlere yol açabilmektedir. Bu yüzden baz istasyonunun yerinin güvenli olması gerekmektedir. Ayrıca üye düğümlerine saldırganlar tarafından ulaşılarak manipüle edilmesi Replay, Flooding, Blackhole gibi atakların ağda gerçekleşmesine yol açmaktadır. Bu düğümlerin konumlarının güvenli olması ağın güvenliği için önemlidir. Ancak her zaman bu durum mümkün olmadığından bunun için düğümlerin haberleşmesinde Message Authentication Code (MAC) ve dijital imza kullanılması ağın güvenliği için önem arz etmektedir [29].

2.3.10. Zaman senkronizasyonu

Zaman senkronizasyonunun gerekliliği düğümlerin enerji kullanımını ideal hale getirerek tasarruf sağlamaktadır. Limitli kaynaklarla çalışan KSA'nın çalışma ömrü kalan enerji miktarları ile doğru orantılıdır. Zaman senkronizasyonu sayesinde belirlenen düğümler arasında belirlenen zaman aralıkları dışında iletişim kesilmektedir. Gecikme zamanı, zaman senkronizasyonunda gerçekleştirilirken bu hesaplamanın dışında tutulmalıdır. Bu nedenle zaman senkronizasyonu iyi ayarlanarak uygulanmalıdır. Aksi halde iletişimde aksaklıklar yaşanmasına neden olabilmektedir [30].

2.3.11. Servis kalitesi

Servis kalitesi, düğümlerin iletişiminin sağlıklı ve düzgün olmasını sağlamaktadır. Aynı kanalda iletişimin çakışması, çakışmadan dolayı paketlerin düşürülmesi, iletilen paketlerin hedef düğümde ya da düğümlerde aşırı yığılmalara sebep olması, iletişimde sıkışmaların oluşması gibi etkenler servis kalitesini düşürmektedir. Bunun sonucunda da iletişimde aksamalar oluşmaktadır. Servis kalitesi ağda harcanan enerjiyi de dolaylı olarak etkilemektedir [31]. Bu yüzden servis kalitesinin kontrolü ağın ömrü, kullanıcılara kaliteli bir iletişimin sağlanması, iletişimin zamanında ve eksiksiz sağlaması için önemlidir.

2.3.12. İleri-Geri gizlilik

Bu özellik kaynak düğümde oluşturulan verinin hedef düğüme ulaşana kadar verinin gizliliğinin korunmasının sağlanması için gereklidir. Kaynak düğümde üretilen veri şifrelenerek hedef düğüme gönderilmek için yol üzerinde bulunan düğümlere iletilmektedir. Şifreleme, yol üzerinde bulunan düğümler tarafından verinin okunmasını önlemektedir [18]. Kısacası yetkisiz düğümlerin veriye ulaşmalarını engellemektedir. Bunun için veriler şifrelenerek gönderilmektedir.

2.4. KSA'da Güvenlik Zafiyetleri ve Saldırı Türleri

KSA altyapı gerektirmeyen, kullanım kolaylığı sağlayan, oto senkron özelliklerinden dolayı farklı alanlarda kabul görmektedir. Ancak bunun yanında bazı güvenlik zafiyetlerini de beraberinde getirmektedir [32]. Bu zafiyetler saldırganlar tarafından kullanmakta ve kimi zaman ağ erişilmez kılınmakta kimi zaman ise ağ manipüle edilmektedir. Saldırılar bazen fiziksel bazen radyo frekansları ile yapılsa da genellikle saldırganın hedef ağa yerleştirdiği zararlı düğümler üzerinden yapılmaktadır. Bu düğümler saldırıda etkin olarak kullanılabilmesi için diğer düğümler açısından cazip hale getirilmektedir. Bunun için ağın ve ağdaki düğümlerin özellikleri iyi bilinmelidir. Bu bilgiler ışığında saldırı etkili ya da etkisiz olmaktadır. Saldırının amacına göre saldırının türü ve şekli değişmektedir. Ancak genel olarak saldırıların amacı;

a. Ağı manipüle etmek ve erişilmez kılmak
b. Ağda gecikmeler sağlayarak etkinliğini ve performansını düşürmek

c. Ağ kaynaklarını tüketmek

d. Ağdaki hassas verileri ele geçirmek

şeklinde sıralanmaktadır. KSA ağları katmanlardan oluşmaktadır. Yapılan bu saldırılar katmanlara göre sınıflandırılabilmektedir. Çizelge 2.1'de saldırı türleri açıklanmış ve katmanlara göre sınıflandırılmıştır.

Çizelge 2.1. KSA saldırı türleri

Sıra No	Katman Adı	Saldırı Türü	Açıklama
1	Fiziksel Katman	Jamming	Radyo frekanslarını karıştırarak ya da keserek yapılan saldırı türüdür.[7]
		Tampering	Fiziksel olarak saldırganın sensör düğümüne müdahale ederek verilerin çalınmasıdır [18].
2	Veri Bağı Katmanı	Collision	Saldırgan tarafından verinin düzgün iletilmemesi kaynakların tüketilmesi için çakışmalar meydana getirilir [18].
		Exhaustion	Kaynakların yoğun tüketerek ya da bant ve düğümleri meşgul ederek yapılır.
		Unfairness	Saldırganın sensör düğümlerini ağa tamamen erişimsiz kılmak yerine kullanımlarını kısıtlamasıdır.
3	Ağ Katmanı	Sinkhole	Zararlı düğümün kendini ağdaki normal düğümlere bazı parametreler bakımından cazip göstererek ağ trafiğini üzerine çekip ağı etkisiz kılmak ya da manipüle etmektir [18].
		Blackhole	Kara delik atağı bir DDoS atak çeşididir. Blackhole atağının çalışma prensibi üzerindeki trafiğin tamamını iletmek yerine üzerine gelen paketleri düşürerek iletişimi tamamen ya da kısmen kopmasını sağlamaktır [12]. Blackhole saldırısı saldırgan düğümün durumuna göre internal ve external olmak üzere ikiye ayrılmaktadır. Internal blackhole ataklarında ağda bulunan ve üzerinde trafik akan normal düğümün dış etkiler ile manipüle edilerek paketleri düşürmesidir. External blackhole saldırılarında ise ağın dışında bulunana bir düğümün sanki ağ içerisineymiş gibi ve cazip düğüm gibi kendini göstererek ağ trafiğini üzerine çekerek paketleri düşürmesidir [32].

Çizelge 2.1. (devam) KSA saldırı türleri

Wormhole	Saldırganın ağa iki zararlı düğüm koyarak çevredeki düğümlere bu iki düğümü cazip ve en hızlı iletim sağlayan düğümler olarak gösterip ağ trafiğini üzerine çekmesidir. Trafiği eline geçiren saldırgan istediği gibi veri kullanıp ağı manipüle edebilir [18].
Sybil	Zararlı düğümün ağa birden fazla farklı ve sahte kimlik ile kendini tanıtıp ağı manipüle etmesidir.
Selective Forwarding	Kendilerine gönderilen her trafiği düşürüp iletmemek yerine bazı trafik paketlerini iletip bazılarını iletmeyerek ağın performansını veetkinliğini azaltır [33].
Flooding	Zararlı düğümün ağdaki diğer düğümlere yeni katılan normal bir düğüm gibi hello paketi göndererek yapılan saldırı çeşitidir. Normal düğümlerden farklı olarak hello paketini sürekli göndererek ağın kaynaklarını tüketmeyi ve erişilmez kılmayı hedefler [32].

3. LİTERATÜR ÇALIŞMALARI

Gerek bilgi sistemleri gerek sensör ağları çalışmaları sonucunda veri setleri elde edilmektedir. Literatürde oldukça farklı bilgi sistemlerine ait ağ trafik veri seti olmasına rağmen sensör ağlar ile ilgili bunu söylemek mümkün değildir. Literatür tarandığında KSA veri seti kullanılarak yapılan çalışmaların da olduğu görülmüştür. Ancak çalışmaların büyük bölümünde veri setlerinin isimlerine, özelliklerine, elde ediliş biçimi gibi bilgilere yer verilmediği yalnızca veri seti analiz sonuçlarına yer verildiği görülmüştür. Bunun yanında farklı çalışmalarda ortak kullanılan veri setinin WSN-DS veri seti olduğu görülmüştür. Literatür incelemesi yapılırken Web of Science, Science Direct, Eric, Elsevier veri tabanları taranmıştır. Özellikle ağ güvenliği çalışmalarında veri setleri ile makine veya derin öğrenme modelleri beraber kullanıldığı için literatür taraması sadece veri seti üzerinden değil aynı zamanda sensör ağlarda makine öğrenmesi ve derin öğrenme ile ilgili çalışmalar üzerinden de gerçekleştirilmiştir. Literatür incelemesi ile ilgili sonuçlara Çizelge 3.1'de yer verilmiştir.

Çizelge 3.1. Literatür özeti

Referans No	Yazar-Yıl	Metot-Model	Veriseti Adı	Saldırı Türü	Platform	Değerlendirme metrikleri ve sonuçları
[33]	Almomani, Al-Kasasbeh, Al-Akhras 2016	ANN MLP Three hidden Layer	WSN-DS	Blackhole Grayhole Scheduling Flooding	NS-2	Accuracy Blackhole: %92,8 Flooding: %99,4 Scheduling: %92,2 Grayhole: %75,6 Normal Case: %99,8
[34]	Ifzarne, Tabbaa, Hadi, Lamghari 2021	SVM Naive Bayes Random Forest Devision Tree	WSN-DS	Blackhole Grayhole Scheduling Flooding	NS-2	Accuracy SVM %89 Naive Bayes %94 Random Forest %94 Devision Tree %94 Model %96
[35]	Garcia, Garrigues, Pous 2016	One Class SVM	Row sound data for 14 days period	Jamming Selective Forwarding	Catalia 3.3	TPR At least %56 FPR Max %5 and %26
[36]	Alrajeh, Khan, Lloret, Loo 2014	ANN	----------	Flooding Routing loop Fake Channel reques	NS-2	Accuracy Flooding %98 Routing loop %95 Fake Channel request %55
[37]	Otoum, Kantarcı, Mouftah 2019	Machine Learning Deep Learning	KDD'99	DoS R2L U2L Probe	-----	Accuracy Machine Learning %99,12 Deep Learning %99,91
[38]	Alshinina, Elleithy 2018	SWSNM CNN Decision Tree ANN SVM	NSL-KDD ----------	Malicious Node Detection	NS-2	Accuracy CNN %87 SWSNM %86,5 Decision Tree %81,5 %

Çizelge 3.1. (devam) Literatür özeti

[39]	Pawar, Anuratha 2021	LSTM	No name	Blackhole Wormhole	Python	Accuracy %90,06
[40]	Bahsi, Nomm, La Torre 2018	Decision Tree	N-BaIoT	DDoS Mirai-Bashlite-Beign	IoT ağları	Accuracy %99,97
[41]	Sokolov, Iliev, Stoyanov 2019	Deep Neural Networks	N-BaIoT and ML Repository	DDoS Mirai-Bashlite	IoT ağları	Accuracy %83
[42]	Nomm, Bahsi 2018	Local Outlier Factor (LOF), One-Class SVM, Isolation Forest (IF)	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	Accuracy %99
[43]	Anthi ve diğerleri 2019	Naive Bayes, Byessian Network, J48, ZeroR, OneR, Simple Logistic, SVM, Multi-Layer Perceptron, Random Forest	----------	Reconnaissance DoS/DDoS MITM Replay Spoofing	IoT ağları	F- Measure %96,2 %90 %98
[44]	Kumar, Lim 2019	RandomForest, K-NN, Gaussian Naive Bayes	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	F1 Score %96
[45]	Possebon ve diğerleri 2019	Ensamble-KNN, DT, MLP	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	Recall %99 Precision %99 Accuracy %99 F1-Score %49
[46]	Meidan ve diğerleri 2017	GBM, Random Forrest, XGBoost	----------	IoT cihaz türü	IoT-PC Ağları	Accuracy %99
[47]	Bai, Liu, Zhang 2019	CNN	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	Accuracy %99,57
[48]	Karanja, Masube, Jeffrey 2019	KNN, Naive Bayes, Random Forest	N-BaIoT	DDoS Mirai-Bashlite	IoT Ağları	KNN %89 Naive Bayes %80 Random Forest %95
[49]	Cvitic, Perakovic, Persa, Botica	KNN, SVM, DT, Random Forest And Artificial Neural Networks	MTC	DDoS	IoT	Accuracy %100
[50]	Liu, Wang, Li, Hao, Feng 2018	CNN	N-BaIoT	Blackhole	OMNET++	PDR %99,9 Residual Energy %88,5
[51]	Anwer, Farooq, Waseemullah 2021	Support Vector Machine (SVM), Gradient Boosted Decision Trees and Random Forest	NSL-KDD	Malicious Trafik	IoT Ağları	Accuracy %85,34
[52]	Okur, Dener 2021	Random Forest	WSN-DS	Blackhole Grayhole Scheduling Flooding	NS-2	Accuracy %99,72
[53]	Okur, Dener 2020	Decision Tree	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	Accuracy %99,95
[54]	Okur, Dener 2022	Random Forest	N-BaIoT	DDoS Mirai-Bashlite	IoT ağları	Accuracy %99,92
[55]	Warzali, Ahmad 2021	Gboost Algorihtms- KNN LR SVM Decision Tree LSTM MLP	WSN-DS	Blackhole Grayhole Scheduling Flooding	NS-2	Accuracy %99,6 F-1 Score %98,8 FPR %0,4 FNR %0,13
[56]	Branitskiy, Kotenko 2016	SVM	NSL-KDD KDD'99	Malicious	IoT ağları	Accuracy KDD %99,96 NSL-KDD %99,6

[33, 34, 52, 55] çalışmalarda saldırı tespiti için farklı öğrenme modelleri kullanılmış olsa da veri seti ve sınıflandırma mantığının aynı olduğu görülmektedir. Almomani, Al-Kasasbeh ve Al-Akhras KSA alanında birçok çalışmada kullanılan WSN-DS veri setinin oluşturmuşlardır. NS-2 ortamında dört farklı saldırı ve bir normal trafik olmak üzere toplam 5 farklı ağ trafiği WEKA aracı ile incelemişlerdir [33]. Ifzarne, Tabbaa, Hadi ve Lamghari yaptıkları çalışmada tespit yöntemleri geliştirmişlerdir. Aynı zamanda elde ettikleri sonuçları geliştirdikleri modelde de kullanmışlardır [34]. Garcia, Garrigues ve Pous 14 günlük ses algılayıcı sensör verilerinde gerçekleşen saldırı ve normal trafikleri incelemişlerdir [35]. Alrajeh, Khan, Lloret ve Loo ürettikleri veri setinin enerji tüketim saldırısı içerdiğini ifade etmişler ve ANN öğrenme modeli ile saldırının tespitini yaptıklarını belirtmişlerdir [36]. Otoum, Kantarcı ve Mouftah makine öğrenmesi ve derin öğrenme modelleri ile iki farklı algoritma geliştirilmişler ve bu algoritmaların doğruluk yüzdelerini ve sonuç alma sürelerini karşılaştırmışlardır [37]. Alshinina ve Elleithy uçtan uca güvenli trafik için gözetimsiz öğrenme ile bir model geliştirmişlerdir. False Pozitive Rate (FPR), gecikme süresi, verimlilik ve enerji kullanım miktarını inceleyerek sonuçları değerlendirmişlerdir [38]. Pawar ve Anuratha Wormhole ve Blackhole saldırılarının tespiti ve engellenmesine yönelik bir çalışma yapmışlardır. Saldırı trafiğinin tespitinde LSTM modeli kullanılmıştır. Bu model ile geliştirilen algoritma sayesinde optimum ve saldırının olmadığı yol belirlenmektedir [39]. Bahsi, Nomm ve La Torre veri seti ile saldırı tespiti yapmışlardır. Saldırı tespitinin yanı sıra özellik azaltma işlemi uygulayarak 115 özelliğe sahip IoT botnet saldırı trafiğini 3 özelliğe indirgemişlerdir. Veri setini karar ağaçları modeliyle inceleyerek trafiği kategorize etmişledir. Geliştirdikleri sistemin ağ ile entegre çalışarak IDS/IPS'lerde kullanılabileceğini belirtmişlerdir [40]. Sokolov, Iliev ve Stoyanov spam içerikleri ve resimleri farklı modeller ile sınıflandırmışlar ve sonuçlarını literatürdeki önceki çalışmalar ile karşılaştırmışlardır [41]. Nomm ve Bahsi IoT cihazlarına ait parametreleri denetimsiz öğrenme modelleri ile inceleyerek zararlı trafikleri sınıflandırmışlardır. Özellik azaltma için entropy işlemini uygulamışlardır. 3 özellik ile SVM modeli ve 5 özellikle Isolation Forest modeli sınıflandırma işleminde en yüksek sonuçları vermiştir [42]. Anthi ve arkadaşları IoT sistemleri için saldırı trafik tespit modeli geliştirmişlerdir. 3 katmandan oluşan bu modelde birinci katmanda normal trafik kategorize edilmiş, ikinci katmanda zararlı trafik kategorize edilmiş, üçüncü katmanda ise saldırı trafiği türüne göre kategorize edilmiştir [43]. Kumar ve Lim IoT botnet saldırı veri setini incelenmişler ve %96 doğruluk yüzdesi ile saldırı tespitini gerçekleştirmişlerdir. Önerdikleri modelin büyük ağ modelleri için dağıtık ve modüler çözümler sunduğu ayrıca modelin IoT

zararlı ağ aktivitelerinin tespiti için geliştirildiği belirtmişlerdir [44]. Possebon ve arkadaşları meta-learning tekniği ile farklı modelleri birleştirerek beraber ve ayrı ayrı kullanımlarını incelemişlerdir. Geliştirilen farklı algoritmalar beraber kullanıldıklarındaki sonuçları ve ayrı ayrı kullanıldıklarındaki sonuçları karşılaştırılmıştır [45]. Meidan ve arkadaşları IoT cihazlarının derin öğrenme modelleri ile inceleyerek marka ve modellerine göre sınıflandırma işlemlerini gerçekleştirmişlerdir. Modelin yetkisiz erişimleri engellemek için kullanılabileceği belirtilmiştir [46]. Bai, Liu ve Zhang 23 özellikle yaptıkları çalışmada veri setini 23x23 boyutunda matris haline getirerek CNN algoritması ile incelemişlerdir. Veri setini 115 özellikten Z Score aracı ile 23 özelliğe düşürmüşlerdir [47]. Karanja, Masube ve Jeffrey veri setini KNN, Random Forest, Naive Bayes modelleri ile analiz etmişlerdir. Sayısal veri setini grinin tonlarında resimlere çevirmişlerdir. Yöntemde düşük matematiksel hesaplamalar ve bağımsız platform özelliklerinin önemi vurgulanmıştır [48]. Cvitic, Perakovic, Persa ve Botica akıllı ev cihazları gibi IoT cihazlarının ürettiği trafik olan MTC ve HTC trafiklerini incelemişlerdir. SHIoT cihazlarının ürettiği DDoS trafiğini MTC kullanılarak %95 doğruluk yüzdesi ile tespit edildiği belirtilmiştir [49]. Liu, Wang, Li, Hao ve Feng etkili yönlendirme modeli ile güven mekanizmasını birleştirmişlerdir. Geliştirilen model ile ağın güvenliği, etkili enerji kullanımı, paket dağıtım oranı gibi başlıklarda yüksek performans sağlandığı belirtilmiştir. [50]. Anwer, Farooq ve Waseemullah NSL-KDD veri setini Random Forest algoritması ile incelemişler ve en yüksek doğruluk yüzdesine %85,34 ile ulaşmışlardır. Yetkisiz uzaktan erişim, DoS, uzaktan süper root yetkisi, port tarama gibi saldırılara sınıflandırma işlemi uygulamışlardır [51]. Okur ve Dener KSA alanından kabul görmüş olan WSN-DS veri setini 7 farklı makine öğrenme modeli ile incelemişlerdir. En yüksek doğruluk yüzdesine %99,72 ile Random Forest modeli ile ulaşmışlardır [52]. Okur ve Dener IoT botnet saldırı trafiğini makine öğrenme modelleri ile incelenmişler ve en yüksek doğruluk yüzdesine %99,95 ile Decision Tree algoritması ile ulaşmışlardır [53]. Okur ve Dener N-BaIoT saldırı veri setini 23 farklı makine öğrenme modeli ile incelenmişler ve en yüksek doğruluk yüzdesine %99,92 ile Random Forest algoritması ile ulaşmışlardır [54]. Okur ve Dener saldırı tespiti için farklı öğrenme modellerini kullanmışlar ve sınıflandırmada en yüksek doğruluk yüzdesine %99,6 ile ulaşmışlardır. Warzali ve Ahmad WSN-DS veri setini farklı öğrenme modelleri ile incelemişlerdir. Sınıflandırma yaparken saldırı türüne göre sınıflandırma yapmışlardır. False Pozitive Rate ve False Negative Rate yüzde sonuçlarına da yer vermişlerdir [55]. Branitskiy ve Kotenko hibrit yaklaşımla bir model geliştirdiklerinden bahsetmişlerdir. Önerdikleri modelde farklı bileşenler olduğu ifade edilmiştir. Bu bileşenlerle beraber SVM öğrenme modelinin de kullanıldığından

bahsedilmiştir. Önerilen modelin birkaç katmandan oluştuğu ve tespit mekanizmalarında kullanılabileceği belirtilmiştir [56]. Literatürde yapılan çalışmalar ile bu tez çalışması karşılaştırıldığında benzerlikler ve farklılıklar olduğu görülmektedir. Ayrıca bu çalışmanın diğer çalışmalardan farklı olarak literatüre yeni katkılarının olduğu düşünülmektedir. Literatür çalışması yapılırken Web of Science, Science Direct, Eric, Elsevier veri tabanları taranmıştır. Çalışmalar incelendiğinde Çizelge 3.1'de görüldüğü gibi KSA alanındaki çalışmaların büyük bir bölümünde WSN-DS veri seti kullanılmıştır. WSN-DS veri setinin kullanılmadığı diğer çalışmalarda ise analiz edilen veri setleri hakkında açıklayıcı ve net bilgi verilmemiştir. Literatürde KSA alanından yapılan veri seti çalışmalarına ait bilgiler Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2. KSA alanında kullanılan veri setleri

Veri Seti Adı	İçerdiği Saldırı Türleri	Veri Setindeki Örnek Sayısı	Veri Setinin Boyutu	Veri Setinin Özellik Sayısı	Veri Setinin Elde Edildiği Ortam
No name	Jamming Selective Forwarding	5 344	------	8	Gerçek Ortam / Castalia
WSN-DS	Blackhole Flooding Grayhole Scheduling	374 661	25,3 MB	23	NS-2
WSN-BFSF	Blackhole Flooding Selective Forwarding	312 106	24,7 MB	16	NS-2

Çizelge 3.2' de yer alan ilk çalışma Garci, Garrigues ve Pous'un yaptıkları çalışmadır. Veri setine herhangi bir isim verilmemiştir. Çalışmada elde edilen veri seti literatürde henüz sadece kendi çalışmalarında kullanılmıştır. Elde ettikleri veri seti gerçek ortamdan sensörler aracılığıyla alınan verilerden oluşmaktadır. Veri seti 14 günlük zaman diliminde üretilen ses verisinden oluşmaktadır. Veri setinin gerçek ortamda üretilmesi bakımından WSN-DS veri setinden ve bu çalışmada elde edilen WSN-BFSF veri setinden ayrılmaktadır. WSN-DS ve WSN-BFSF veri setleri benzetim ortamından elde edilmiştir. Ses verisinden oluşan (isimsiz) veri seti içerdiği saldırı (Jamming, Selective Forwarding) sayısı bakımından WSN-DS ve WSN-BFSF veri setlerinde bulunan saldırı sayılarından daha azdır. Literatür incelemeleri sonucunda KSA alanında veri seti sayısının sınırlı olduğu görülmektedir. Bu çalışmada elde edilen WSN-BFSF veri seti bu alana ve bu alanda çalışan geliştiricilere katkı sağlayacağı düşünülmektedir. KSA alanında yapılan çalışmaların büyük bir bölümünde kullanılan WSN-DS veri seti Blackhole, Grayhole, Flooding ve Scheduling saldırılarını içermektedir. WSN-

BFSF veri seti Blackhole, Flooding ve Selective Forwarding saldırılarını içermektedir. WSN-DS veri setinden farklı olarak Selective Forwarding saldırısına ait verileri de içermektedir. Bu bakımdan Selective Forwarding saldırı verisini analiz etmek isteyen geliştiricilerin çalışmalarına ve literatüre katkı sağlayacağı düşünülmektedir. Literatürde bulunan mevcut WSN-DS veri setinin farklı çalışmalarda yaklaşık tüm öğrenme modelleri ile incelendiği görülmektedir. Bu bakımdan literatürde modeller ile incelenmek için yeni veri setlerine ihtiyaç duyulmaktadır. Bu açından da literatüre katkı sağlayacağı düşünülmektedir. WSN-BFSF veri seti 16 özellikten oluşmaktadır. WSN-BFSF'de kullanılan özellikler literatürdeki diğer veri setlerinde kullanılan özelliklerden farklıdır. Bu yönüyle de diğer çalışmalardan ayrılmaktadır.

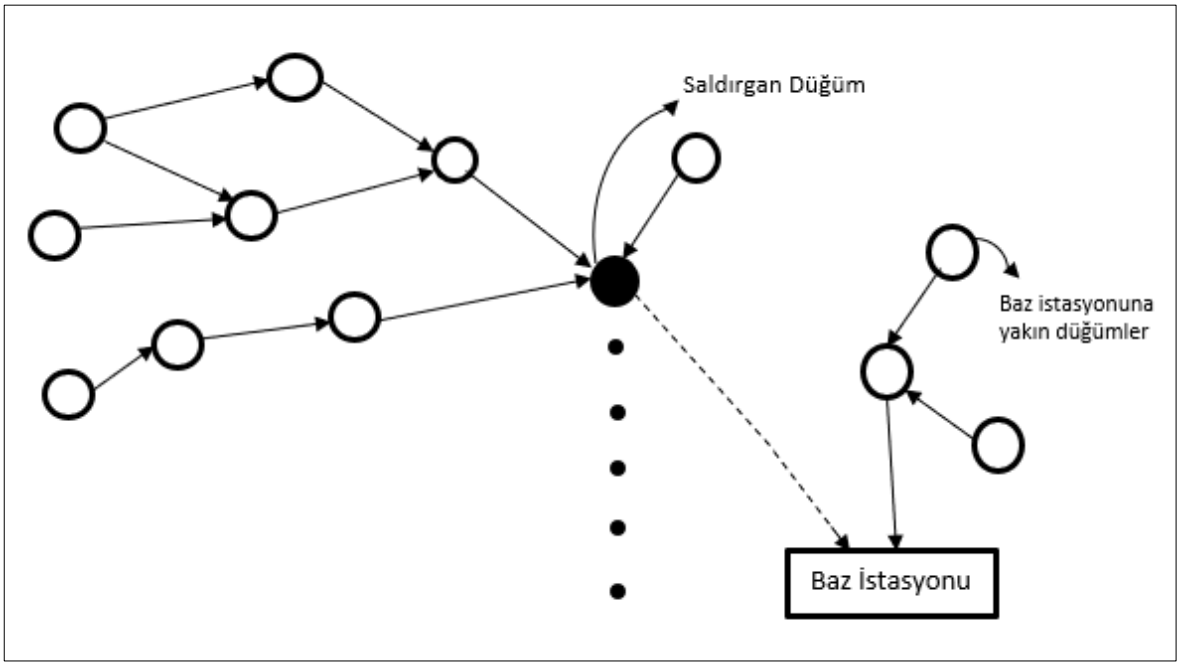
4. ATAK MODELLERİ

KSA günlük hayatta birçok alanda kullanılmasından dolayı farklı gizlilik seviyelerinde bilgiler taşımaktadır. Bu bilgiler KSA'yı saldırganların hedefi haline getirmektedir. KSA'ya gerçekleştirilen saldırılar farklı şekillerde sınıflandırılabilirler. Çalışmada gerçekleştirilen Blackhole, Flooding, Selective Forwarding, saldırılar ağ katman saldırılarına örnek olmakla birlikte aynı zamanda DDoS saldırılarına da örnektir. Ancak DoS ve DDoS saldırılarını birbirinden ayıran en önemli özellik, isminden de anlaşılacağı gibi birincisi servis reddi (DoS) diğeri dağıtık servis reddi (DDoS) saldırılarıdır [57]. İkisinde de amaç hedefi erişilmez kılmak ve kullanıcıların hedefe ulaşmasını engellemektir. İlkinde bu atağı tek saldırgan gerçekleştirirken, ikincisinde birden çok saldırgan gerçekleştirmektedir. Ağa dahil olan cihaz sayısında ivmeli artış bu saldırının cazibesini ve etkisini arttırmaktadır. Özellikle sonuçları bakımından her sensör bir bilgisayar gibi düşünüldüğünde sensör ağlar çalıştığında büyük bir ağ trafiği oluşmakta ve bu trafik saldırganlar tarafından kötüye kullanılmaktadır. Son yıllarda DDoS saldırısının kullanımı zararlı yazılımlar ve bu yazılımları haberli-habersiz kullanan cihazların manipüle edilmesi ile artmıştır. DDoS, DoS saldırısına göre daha senkron ve zahmetli olsa da daha etkili ve zararlı sonuçlar meydana getirmektedir [52]. Aşağıda belirtilen saldırı türleri DoS saldırılarında da DDoS saldırında da kullanılmaktadır. Saldırıların gerçekleştirilme şekilleri farklı olsa da amaçları aynıdır. KSA'larda DoS saldırılarına örnek olarak Blackhole, Flooding, Selective Forwarding, Grayhole saldırıları gösterilebilmektedir.

4.1. Blackhole Saldırısı

Blackhole saldırıları ağda gerçekleşen iletişimin bir ya da birkaç düğüm tarafından kendilerini düğüm güven değeri, kalan enerji miktarı, en kısa yol gibi parametreler açısından cazip göstererek iletişimi üzerlerine çekip trafiği kısmen ya da tamamen kesmek için yapılan saldırılardır [58]. İletişim yolu üzerinde bulunana zararlı düğümlerin kendilerini cazip düğüm gibi göstermeye ihtiyaçları yoktur. İletişim kendileri üzerinden gerçekleştiği için istedikleri an atağı başlatabilmektedirler. Blackhole saldırısını gerçekleştirecek olan düğümlerin diğer tüm düğümler tarafından ulaşılabilir olması önem arz etmektedir. Zararlı düğümler RREQ istek paketlerine ideal ve normal bir düğümmüş gibi cevap vererek kendilerini gizlemektedir. İstek paketlerini gönderen düğümler aldıkları cevap ışığında bu zararlı düğümleri iletişim gerektiğinde normal ara düğüm olarak

kullanabilmektedir. Blackhole saldırısı çok adımlı iletişim modellerinde kullanılmaktadır. Bu iletişimlerde saldırgan düğüm hedefe gönderilen paketi iletmek yerine üzerine gelen paketleri düşürerek iletişimin tamamen ya da kısmen kopmasına neden olmaktadır [2]. Zararlı düğüm; paket içeriğine, gönderen düğümün numarasına, hedef düğümün numarasına, iletişim zamanına, belirlenen gruplara göre paket trafiğini düşürmektedir. Blackhole saldırısı bir DDoS atak çeşididir. Blackhole saldırısı saldırgan düğümün durumuna göre internal ve external olmak üzere ikiye ayrılmaktadır. Internal Blackhole saldırılarında ağda bulunan ve üzerinden trafik geçen normal düğümlerin dış etkiler ile manipüle edilerek zararlı düğüm haline getirilip gelen trafik paketlerini düşürmektedir. External Blackhole saldırılarında ise ağın dışında bulunan bir düğüm sanki ağ içerisindeymiş gibi ve cazip düğüm gibi kendini göstererek ağ trafiğini üzerine çekerek paketleri düşürmektedir [20]. Çalışmadaki saldırı türü internal atak türüne bir örnektir. Şekil 4.1'de Blackhole saldırı modeli gösterilmiştir.



Şekil 4.1. Blackhole saldırı modeli

Şekil 4.1'de görüldüğü gibi Blackhole saldırısını gerçekleştiren düğüm saldırgan düğüm olarak adlandırılmıştır. Saldırgan düğüm kendisine iletilmek üzere gelen paketleri iletmeyerek ağ trafik akışında kopmalara neden olmaktadır. Saldırgan düğüme kadar veri trafiği normal bir şekilde gerçekleşmektedir. Normal veri trafik akışı siyah oklarla gösterilmiştir. Saldırgan düğüm kendi grubundaki düğümlerden gelen paketleri kesikli

çizgilerle gösterilen doğrultuda iletmesi gerekirken iletmemekte ya da düşürmektedir. Blackhole saldırı modelinin çalışma algoritması psuedo kodlar ile şekil 4.2'de gösterilmiştir.

```
SD→ Sensör düğümü
ZD → Zararlı düğüm
AÜD →Ağ üye düğümü
MAÜD →Muhtemel ağ üye düğümü
BD→Bilinmeyen düğüm
DN→Düğüm numarası
P→Paket düşürme oranını hesaplayan fonksiyon
X → Düğüm numarası (0 ≤ x ≤199 olması beklenmektedir.)
IF (DNX, 0 ≤ x ≤199)
        SDx=MAÜD
        IF(P(SDx) ≥ 0.9)
                SDx=ZD
        ELSE
                SDx=AÜD
ELSE
        SDx =BD
        IF(P(SDx) ≥ 0.9)
                SDx=ZD
Gelen tüm paketler düşürülmektedir.
```

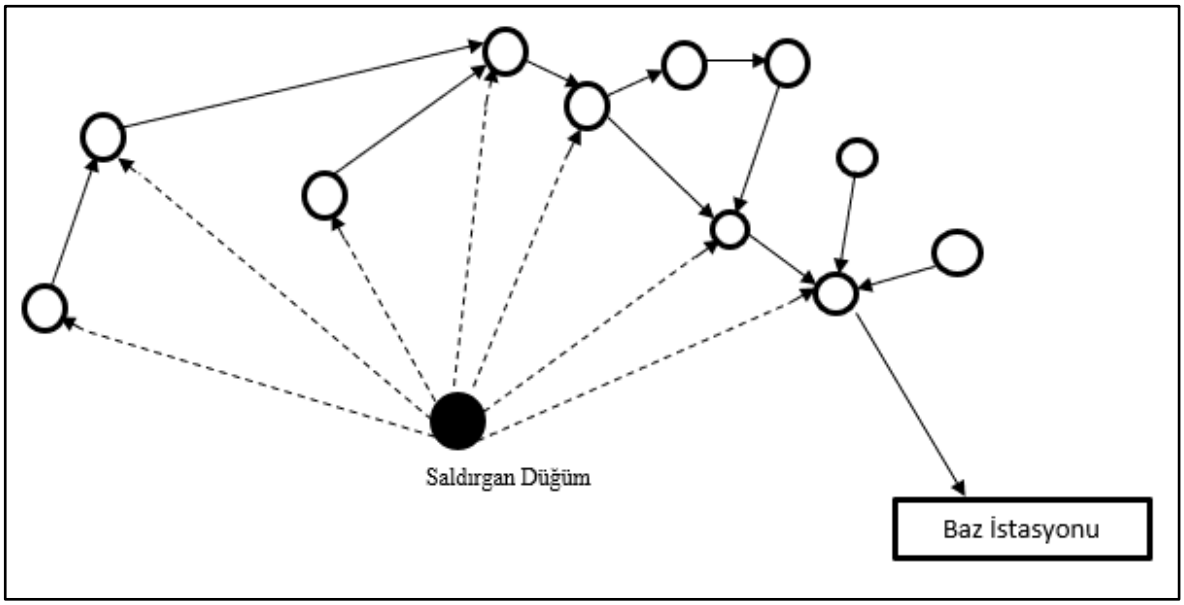
Şekil 4.2. Blackhole saldırısının psuedo kodu

Şekil 4.2'de Blackhole saldırısını yapan düğüm ile ağa ait normal düğümün ayrımının nasıl yapıldığı anlatılmaktadır. Öncelikle düğümün ağa ait olup olmadığı tespit edilmektedir. Tespit yapılırken IF yapısı ile düğüm numarası birlikte kullanılmaktadır. Uygulama bölümünde ağın yapısı hakkında detaylı bilgi verilmiştir. Ağ 200 düğümden oluşmaktadır. Bu yüzden düğüm numaraları 0-199 (numaralar dahil) aralığındaki sayılardan belirlenip atanmıştır. Düğümler incelenirken önce düğüm numaralarına bakılıp ağa ait olup olmadıkları tespit edilmektedir. Düğüm numarası bu sayı aralığı dışında ise bilinmeyen düğüm şeklinde tanımlanmaktadır. Bilinmeyen düğümün sonradan ağa dahil olan yeni düğüm mü yoksa zararlı düğüm mü olup olmadığı paket düşürme oranı sonucuna bakılarak netleştirilmektedir. Bunun için bir P fonksiyonu tanımlanmıştır. Tanımlanan P fonksiyonu; iletilmek üzere gelen paketlerden düğümün düşürdüğü paket sayısının ilettiği paket sayısına oranını hesaplamaktadır. P değeri 1'e çok yakınsa Blackhole saldırısı yapan zararlı düğüm olarak belirlenmektedir. Paket düşürme oranını (P fonksiyonu) gelen paket miktarının iletilen paket miktarından çıkarılıp sonucun gelen paket miktarına bölünmesi ile

hesaplanmaktadır. Diğer taraftan düğüm numarasına bakılarak ağa ait olduğu tespit edilen düğümün manipüle edilip edilmediğini anlamak için ikinci bir test uygulanmaktadır. P fonksiyonu ile hesaplanan paket düşürme oranı 1'e çok yakınsa düğümün saldırgan bir düğüm olduğu ve yaptığı saldırının Blackhole saldırısı olduğu sonucu çıkarılmaktadır.

4.2. Flooding Saldırısı

Ağdaki düğümler iletişime geçmeden önce hangi yoldan nasıl birbirleriyle haberleşeceklerini belirlemek için tanışma ve istek paketleri göndermektedir [59]. Ancak bu sadece ağ iletişimin başlangıcında olmakta ve ağda ciddi bir trafiğe sebep olmaktadır. KSA'ın kurulumunda sensörler arası iletişim senkronizasyonu sağlanması için istek paketleri düğümden diğer tüm düğümlere gönderilmektedir. Diğer taraftan ağa yeni dâhil olan düğümlerde iletişim başlamadan önce istek paketlerine ihtiyaç duymaktadırlar. Gönderilen bu istek paketleri iletişimi başlatmak için gerekmektedir. Bu paketlerden elde edilen bilgiler ışığında tablolar oluşturularak bu tablolara göre yön-güzergâh belirlenmektedir. Ancak bu paketler ağa büyük yük getirmektedir [60]. Amacına uygun kullanıldığında bile ağa yük olurken eğer bu durum iletişim süresi boyunca sürekli olduğunda hem düğümler arasında gerçek iletişimde beklemelere-düşmelere-yavaşlamalara hem ağ trafik yoğunluğunda büyük artışlara hem de sensör enerjilerinin çok kısa zamanda tükenmesine sebep olmaktadır. Saldırgan düğüm ya da düğümler bu durumdan yararlanmaktadır. Saldırgan düğümün normal düğümlerden farklı olarak istek paketlerini rastgele zamanlarda ya da sürekli olarak göndererek ağın kaynaklarını tüketmeyi ve ağı erişilmez kılmayı hedeflemektedir [8]. Şekil 4.3'te Flooding saldırı modeli gösterilmektedir.



Şekil 4.3. Flooding saldırı modeli

Şekil 4.3'te görüldüğü gibi Flooding saldırısını gerçekleştiren düğüm saldırgan düğüm olarak adlandırılmaktadır. Saldırgan düğüm RREQ paketlerini kesikli çizgilerle gösterilen doğrultuda düğümlere göndermektedir. Saldırının gerçekleştirilmesi için RREQ paketleri amacı dışında kullanılmaktadır. Herhangi bir haberleşme amacı ile gönderilmeyen bu istek paketleri ağda trafik yoğunluğuna, trafik gecikmelerine ve bunlara bağlı olarak paket düşürmelerine neden olmaktadır. Bu saldırı ulaşılabilirliği ihlal etmek için gerçekleştirilmektedir. Flooding saldırısının psuedo kodlarına Şekil 4.4'te yer verilmiştir.

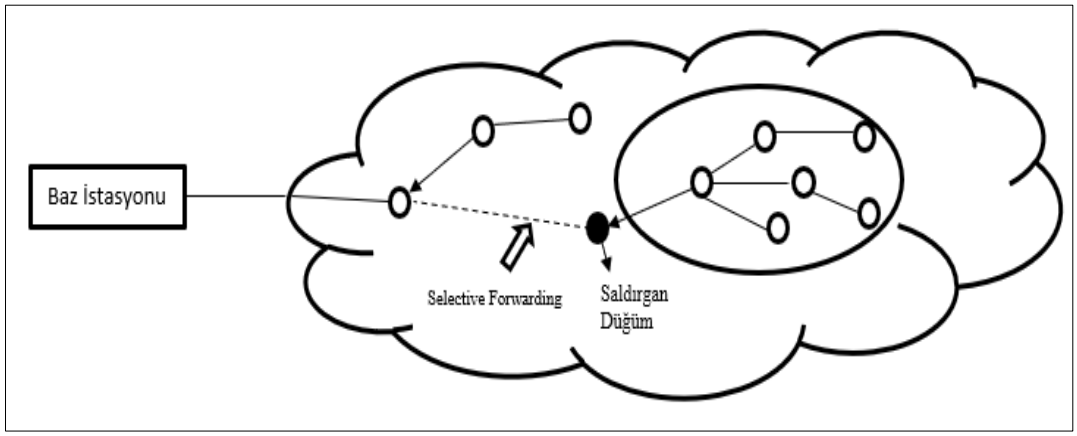
```
SD→ Sensör düğümü
ZD → Zararlı düğüm
AÜD →Ağ üye düğümü
MAÜD →Muhtemel ağ üye düğümü
DN→Düğüm numarası
ZSON → Son gönderilen RREQ paket zamanı
ZILK → İlk gönderilen RREQ paket zamanı
DZ→ TSON – TILK
G→RREQ paketlerinin gönderim zaman aralığı (ağ ortalaması)
X→ Düğüm numarası (0 ≤ x ≤199 olması beklenmektedir.)
IF (DNX, 0 ≤ x ≤199)
        SDX=MAÜD
        IF(G<<DZ)
                SDx=ZD
        ELSE
                SDx=AÜD
ELSE
        SDX =BD
        IF(G<<DZ)
                SDx=ZD
```

Şekil 4.4. Flooding saldırısının psuedo kodu

Şekil 4.4'te Flooding saldırısını yapan düğüm ile ağa ait normal düğümün ayrımının nasıl yapıldığı anlatılmaktadır. Öncelikle düğüm numarası üzerinden ağ üye düğümü olup olmadığı, üye düğüm ise manipüle edilip edilmediği zaman değişkeni kullanılarak tespit edilmektedir. Düğümün ağa ait olup olmadığı tespiti yapılırken IF yapısı ile düğüm numarası birlikte kullanılmaktadır. Ağ 200 düğümden oluşmaktadır. Bu yüzden düğüm numaraları 0-199 (numaralar dahil) aralığındaki sayılardan belirlenip atanmıştır. Düğümler incelenirken önce düğüm numaralarına bakılıp ağa ait olup olmadıkları tespit edilmektedir. Düğüm numarası bu sayı aralığı dışında ise bilinmeyen düğüm şeklinde tanımlanmaktadır. Bilinmeyen düğümün RREQ paket gönderme zaman sıklığı DZ fonksiyonu ile hesaplanmaktadır. Elde edilen sonuç G (ağdaki düğümlerin ortalama RREQ paket gönderme zaman aralığı) değeri ile karşılaştırılarak Flooding saldırısı yapan bir düğüm olup olmadığı tespit edilmektedir. Diğer taraftan düğüm numarası bakılarak ağa ait olduğu tespit edilen düğümlerinde ikinci bir test ile manipüle edilip edilmediğini düğümlerin DZ fonksiyon sonucu ile G değeri ile karşılaştırılarak anlaşılmaktadır.

3.3. Selective Forwarding Saldırısı

Selective Forwarding saldırısı ilk kez Karlof ve Wagner tarafından saldırının çerçevesi çizilerek tanımlanmıştır. Bazı geliştiriciler tarafından çalışma modeli Grayhole saldırısına benzediği için Grayhole saldırısı şeklinde de adlandırılmaktadır [61]. Basit tanımı seçici geçirgen bir saldırı türüdür. Selective Forwarding saldırısı kendisine iletilmesi için gelen tüm paketleri iletmek yerine paketlerin içeriği, gönderim zamanı, gönderen düğümün numarası, hedef düğümün numarası gibi değerleri baz alarak paketlerin bir bölümünü iletmeyi reddeden ya da düşüren bir bölümünü ise ileten saldırı türüdür [21]. Bu durum trafikten etkilenen (kaynak-hedef) düğümler için DoS saldırısı anlamına gelmektedir [62]. Bu yüzden DoS saldırı türü olarak bilinmektedir. Bu saldırıyı diğer saldırılardan ayıran; zararlı düğüm (saldırgan) tarafından belirlenen tekil ya da çoğul düğümlerden gelen paketlerin düşürülmesi ya da iletilmemesidir. Selective Forwarding saldırılarını gerçekleştiren zararlı düğümler çok adımlı iletişimlerde yer alarak ara düğüm görevini üstlenmektedirler. Böylece kendilerine iletilmesi için gelen paketleri manipüle edebilmektedirler. Selective Forwarding saldırılarının bir diğer amacı gelen paketleri bekleterek iletmeleridir. Burada amaç paketleri zamanında göndermeyip sonradan göndererek ağ yönlendirmesinde karışıklığa yol açmaktır. Zararlı düğüm üzerinden bekletilerek hedef düğüme gönderilen paketler hedef düğümde bilgi kirliliğine yol açmaktadır. Bu saldırı modelinde verinin çalınması ya da değiştirilmesi amaçlanmamaktadır. Bu saldırıda iletişim kısmen koparılarak düzgün ve anlaşılır bir iletişimin engellenmesi hedeflenmektedir. Şekil 4.5'te Selective Forwarding saldırı modeline yer verilmektedir. Şekilde 4.5'te gösterildiği üzere baz istasyonuna en yakın düğüm üzerinden veri baz istasyonuna doğrudan aktarılmaktadır. Ancak daha uzak noktadaki düğümler paket gönderirken çok adımlı iletişim gerçekleştirmektedirler. Selective Forwarding saldırısını gerçekleştirecek olan düğüm saldırgan düğüm şeklinde adlandırılmıştır. Küme içindeki düğümler iletişimlerini saldırgan düğüm üzerinden gerçekleştirmektedir. İletişimdeki bu durum saldırgan düğüm tarafından kullanılmaktadır. Şekilde 4.5'te saldırgan düğüm kendi grubundaki trafiğin bir kısmını iletirken bir kısmını düşürerek ya da iletmeyerek (kesikli çizgilerle gösterilen) iletişimde kopmalara neden olmaktadır. Modelde normal trafikler çizgilerle saldırı trafiği kesikli çizgiler gösterilmiştir. Saldırgan düğüm yüzünden baz istasyonuna eksik veriler iletilmektedir. İletilen veriler eksik olduğu için kullanıcı için anlam ifade etmemektedir.



Şekil 4.5. Selective Forwarding saldırı modeli

Diğer taraftan Selective Forwarding saldırısından dolayı baz istasyonuna iletilen verilerin hangisinin güncel hangisinin eski olduğu konusunda karmaşalar yaşanmaktadır. Saldırının temel amacı veri bütünlüğünü ihlal etmeyi hedeflemektedir. Saldırının model ve çalışma mantığı böyleyken algoritma tarafını anlatmak için psuedo kod kullanılmıştır. Selective Forwarding saldırısına ait psuedo kodları Şekil 4.6'da gösterilmiştir.

SD→ Sensör düğümü
ZD → Zararlı düğüm
AÜD →Ağ üye düğümü
MAÜD →Muhtemel ağ üye düğümü
BD→Bilinmeyen düğüm
DN→Düğüm numarası
R→Paket düşürme oranını hesaplayan fonksiyon
K→Ağın ortalama paket düşürme oranı
X→ Düğüm numarası ($0 \leq x \leq 199$ olması beklenmektedir.)

```
IF (DNX, 0 ≤ x ≤199)
        SDX=MAÜD
        IF (R(SDX)>K)
                SDx=ZD
        ELSE
                SDx=AÜD
ELSE
        SDX =BD
        IF (R(SDX)>K)
                SDX=ZD

Gelen paketler belirli oranlarda düşürülmektedir.
```

Şekil 4.6. Selective Forwarding saldırısının psuedo kodu

Şekil 4.6'da Selective Forwarding saldırısını yapan düğüm ile ağa ait normal düğümün ayrımının nasıl yapıldığı anlatılmaktadır. Tıpkı diğer saldırılarda olduğu gibi bu saldırıda da öncelikle düğümün ağa ait olup olmadığı düğüm numarası karşılaştırılarak tespit edilmektedir. Düğüm numarası ağa ait değilse bilinmeyen düğüm olarak adlandırılmaktadır. Bilinmeyen düğüm olarak adlandırılan düğüme ikinci test uygulanmaktadır. Düğümlerin paket düşürme oranını belirleyen fonksiyon (R) oluşturulmuştur. Paket düşürme oranı gelen paket sayısının iletilen paket sayısından çıkarılarak elde edilen sonucun gelen paket sayısına bölünmesi ile elde edilmektedir. İlgili düğümün R fonksiyonu ile değerlendirilip paket düşürme oranı belirlenerek, K (ağın ortalama paket düşürme oranı) değeri ile karşılaştırılmaktadır. R fonksiyon sonucu K değerinden büyük ise zararlı düğüm olarak adlandırılmaktadır. Diğer taraftan düğüm numarası ağa ait olduğu tespit edilen düğüme de R fonksiyonu uygulanarak sonucu K değeri ile karşılaştırılmaktadır. Eğer düğümün R değeri K değerinden büyükse ağa ait düğümün manipüle edildiği anlaşılmaktadır.

5. YAPAY ZEKÂ VE TÜRLERİ

5.1. Yapay Zeka

Günümüzde kullanılan cihaz ve teknolojilerin artması ile bunların kullanımı ve yönetilmesi büyük iş gücü gerektirmektedir. Kullanıcılar günlük hayattaki uzun ve çaba gerektiren problem ve analizlerini yaparken geçmişte kazandıkları edinimler doğrultusunda bazen doğru bazen yanlış kararlar verebilmektedirler. Son yıllarda özellikle popülaritesi artan yapay zekâ, verilerden bir edinim çıkaran ve bu edinimden bir karar, çözüm, sonuç çıkaran bilim dalıdır. Yapay zeka; geçmişteki problemler, çözümler, sonuçlar, ilişkili veriler ile kendisinden yorumlamasını istenen veri ile ilişki kurup en yakın ve realist sonucu üreten bir teknolojidir [63]. Yapay zeka uzman sistemler, yapay sinir ağları, genetik algoritmalar, bulanık önermeler mantığı gibi yaklaşım ve modellerden faydalanarak sonuçlar üretir. Yapay sinir ağları modeli yapay zeka öğrenmelerinde yaygın olarak kullanılan ve bir çok makine öğrenmelerinin de temelini oluşturan modeldir.

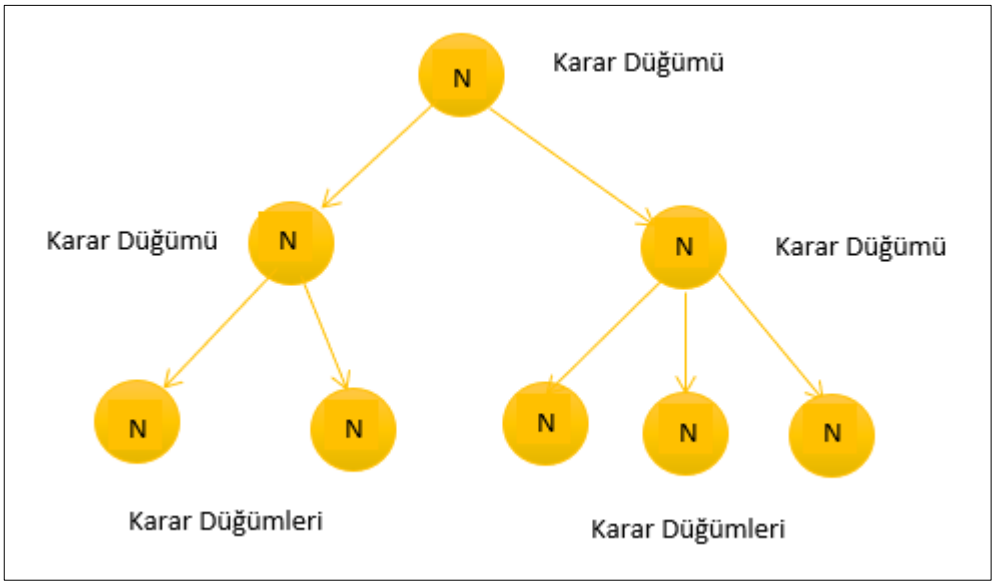
5.2. Makine Öğrenmesi

Yapay zekâ çok geniş bir alanı kapsamaktadır. Bu alana giren makine öğrenmesi son yıllarda popülaritesi artan ve birçok alanda uygulanabilen yapay zekanın bir alt dalıdır. Makine öğrenmesi matematiksel ve istatistiksel işlemler ile veriler üzerinden çıkarımlar yaparak tahminlerde bulunan sistemlerin bilgisayarlar ile modellenmesidir [63]. Makine öğrenmesinde önceden hazırlanan verilerin ön işlemden geçirilerek durulaştırılması sağlanır. Ön işlemden geçirilen veriler için özellik çıkarımı yapılır. Bundan sonraki aşamada öğrenmenin türüne göre kümele ya da sınıflandırma yöntemi ile öğrenme gerçekleştirilir. Makine öğrenmesi gözetimli öğrenme (Supervised Learning), gözetimsiz öğrenme (Unsupervised Learning) ve destekli öğrenme (Reinforcement Learning) olmak üzere üçe ayrılır. Gözetimli öğrenme sınıflandırma temelli çalışır. Gözetimsiz öğrenme ise kümeleme temelli çalışır. Gözetimli öğrenmede girdi verileri etiketlidir. Öğrenmenin sonucunun ne olması gerektiği bellidir. Kısaca girdilerin ve sonuçlarının belli olduğu bir öğrenme modelidir [64]. Gözetimsiz öğrenmede girdiler etiketsiz olup girdilerin nasıl kümeleneceği ve sonuçların nasıl çıkarılacağı kullanılan gözetimsiz öğrenme algoritmasına bağlıdır. Bu belirleme işlemini kullanılan gözetimsiz öğrenme algoritma tarafından yapılır. Destekli

öğrenme biçimi tamamen ortamın gereklerine göre öğrenmenin gerçekleştirilmesidir. Ortamın ve şartların durumuna göre sonucu belirleyen değişkenlerin ağırlıkları yüksek değer ile değerlendirilir. Test edilerek geri dönüş alınır. İstenilen sonuca ulaşıncaya kadar bu geri besleme işlemi devam eder. Öğrenme istenilen sonuca varmayı hedefler ve o yönde seçim ve değerlendirme yapar. Değer tabanlı, politika tabanlı, model tabanlı çalışma şeklinde üç yaklaşım bulunmaktadır [35]. Derin öğrenme bir makine öğrenme modeli olmasından dolayı derin öğrenmede gözetimli, gözetimsiz ve takviyeli öğrenme olmak üzere gerçekleştirilebilir.

5.2.1. Decision Tree

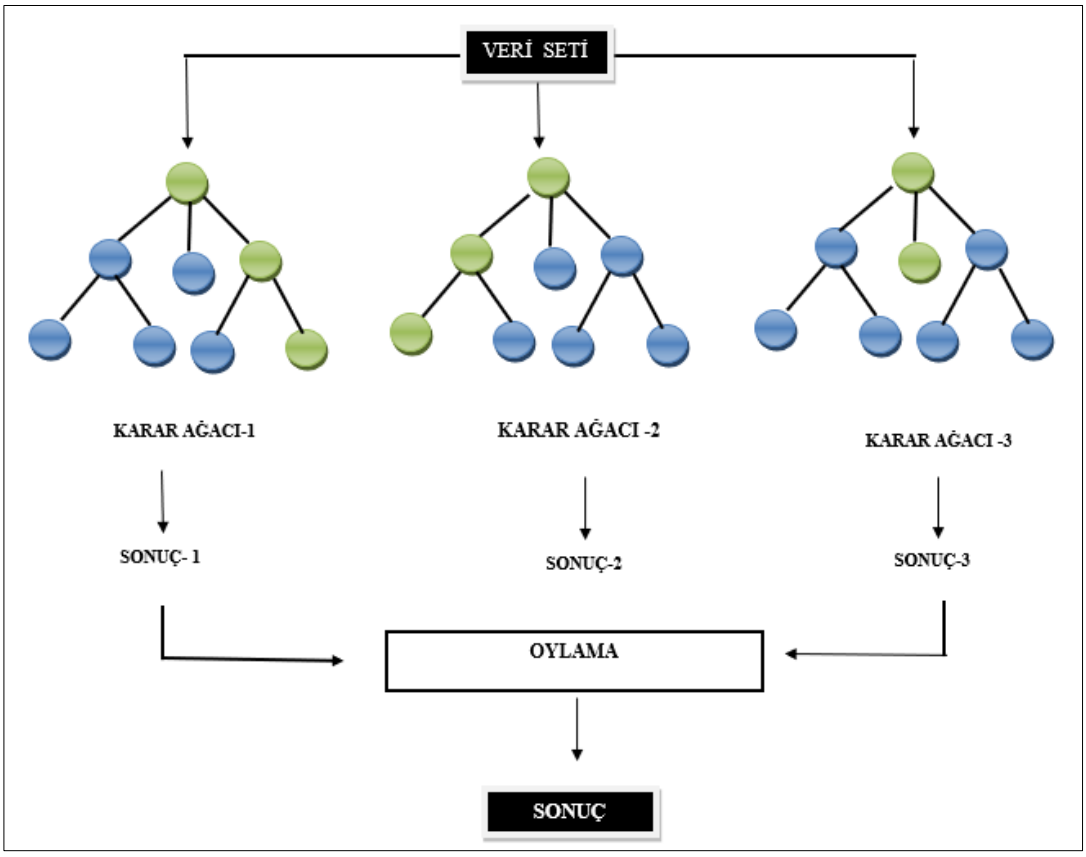
Decision Tree algoritması makine öğrenmesinde kullanılan büyük boyuttaki verileri küçük boyutlara belirlenen karar ışığında sınıflandıran algoritmadır. Sınıflandırma yapısı ağaç dallarına benzediğinden ve sınıflandırmanın karar mekanizmasına göre yapıldığından bu şekilde isimlendirilmektedir. Önceki modelinde J48 algoritması adıyla da bilinmektedir. Algoritmada verinin içeriğine göre çok sayıda kategori oluşturulabilir. Bu durum avantaj olduğu gibi karmaşıklığa da neden olabilmektedir [60]. Algoritmanın ağaç yapısı görselleştirilebilir. Weka platformunda Decision Tree algoritması çok büyük veri boyutunda hata verebilmektedir. Ayrıca öğrenmede ezber yaşanabilmektedir. Bu durumda elde edilen sınıflandırmanın doğruluk yüzdesi yüksek olsa bile bir anlam ifade etmemektedir. Bunu engellemek için parametre kısıtlamasına gidilebilmektedir. Decision Tree algoritma modeli Şekil 5.1'de gösterilmiştir.



Şekil.5.1. Decision Tree algoritma şeması

5.2.2. Random Forest

Random Forest modeli Decision Tree algoritmalarının beraber kullanıldığı ve oylama mantığıyla çalışan bir makine öğrenme modelidir. Decision Tree modelinde bulunan eksiklik nedeniyle bu model avantajlı duruma geçmektedir. Decision Tree modelinde veri işlenirken ezber ile öğrenme dezavantajlı durumu bu modelde giderilmiştir. Yukarıda belirtildiği gibi oylama mantığı ezberlemenin önüne geçmek için geliştirilmiştir. Farklı sayılardaki Decision Tree algoritmaların her birinin sonuçlarına bakılarak en yüksek oy alan ağacın sonucu kullanılmaktadır. Algoritma ile birden fazla ve farklı boyutlardaki veri seti ile birden fazla Decision Tree ile öğrenme gerçekleştirilip sağlaması yapılarak ezberlemenin önüne geçilmektedir. Farklı sayıda kullanılan Decision Tree algoritması ve bu algoritmaların değerlendirme sonuçlarına oylama mantığıyla sonuç üretmesi nedeni ile Decision Tree modelinden daha kompleks bir yapıya sahiptir [60]. Random Forest modeli Şekil 5.2'de gösterilmiştir.



5.2. Random Forest algoritma şeması

5.2.3. Naive Bayes

Çalışma prensibi olasılık üzerine kurulu olan bir modeldir. Temeli Bayes teoremine dayanmaktadır [63]. Eğer veri setinin boyutu az ise ve öğrenme içinde az veri ayrılmış ise kullanılması ideal bir algoritmadır. Algoritma her veri örneği için aitlik olasılığı hesaplamakta ve en yüksek olasılık değerine göre sınıflandırmaktadır. Dengesiz veri dağılımına sahip setlerde de kullanılabilmektedir. A değeri sınıf ifade ederken, B değeri veriyi ifade etmektedir.

$$P(A \mid \mathrm{B}) = \frac{P\ (B \mid \mathrm{A}).\mathrm{P}(\mathrm{A})}{P(B)} \tag{5.1}$$

5.2.3. Logistic Regression

Logistic regression, bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan bir veri kümesini analiz etmek için kullanılan istatistiksel bir yöntemdir. İstatistik bir yöntem olan LR sonucu etkileyen birden fazla etkenin ya da değişkenin olduğu verileri incelemektedir. Logistic Regression'ın amacı bağımlı değişken ile bağımsız değişken arasında ilişkiyi bulabilecek en uygun modeli bulmaktır. Logistic Regression Liner Regression'a göre daha karmaşık sınıflandırmalarda tercih edilmemektedir. Linear Regression ile benzer özellikleri olsa da Linear Regression'ın daha gelişmiş bir modeldir. Temelinde sigmoid fonksiyon yatmaktadır. Sigmoid fonksiyon çıktı değerleri 1-0 olduğu düşünüldüğünde benzerlik gösterdiği görülebilmektedir.

5.2.4. Derin Öğrenme

Derin öğrenme yaklaşımı makine öğrenmesinin bir alt dalı olup bu çalışmada kullanılan bir yaklaşımdır. Derin öğrenme, bir makine öğrenmesi yöntemi olmasından dolayı derin öğrenmede de gözetimli ve gözetimsiz öğrenme gerçekleşmektedir. Derin Öğrenme modelleri yapay sinir ağlarını temel alan ve yapısı sinir ağları mimarisine sahiptir [63].

Eğitim seti öğrenmenin en önemli kısmıdır. Öğrenme işleminin belirleyicisidir. Bu veri seti ile öğrenme sağlanır. Eğer gözetimli öğrenme gerçekleştirilmesi isteniyorsa eğitim setleri etiketli olmalıdır. Buradaki yaklaşım istenilen girdilerle istenilen sonucun alınmak istenmesidir. Doğrulama veri seti öğrenmenin gerçekleşmesine yardım eder. Optimal

ağırlığı elde etmek için fonksiyon parametrelerinin belirlenmesine yardım eder. Test seti, öğrenmenin ne derece başarılı olduğunun görülmesini sağlar. Öğrenme işlemi gerçekleştirilmeden önce eldeki veri seti eğitim ve test seti olarak ikiye ayrılır.

Doğrulama veri seti test setindeki verileri kullanır. Sinir ağları derin öğrenmenin temel aldığı bir modeldir. Canlıların sinir sistemi yapısından esinlenerek geliştirilmiş bir modeldir. Tıpkı canlıların sinir sistemin için nöronlar ne kadar önemli ve belirleyici ise sinir ağları modelinde de nöronlar(düğümler) o kadar önemlidir. Bu sinir ağları giriş katmanı, çıkış katmanı ya da giriş katmanı gizli katman ve çıkış katmanı şeklinde farklı sayıda katmanlardan oluşabilir [64]. Giriş katmanı ile çıkış katmanı arasında kalan katmanlara gizli katman denilmektedir. Ara ve çıkış katmanlardaki nöronların bir ya da birden fazla önceki nöronlardan gelen girdileri vardır. Bu nöronlar belirlenen aktivasyon fonksiyonlarına göre girdileri işleme tabi tutup sonuç üretirler. Basit olarak matematiksel gösterimi;

$$Z = \sum (\text{Input}) * (\text{Weight}) + \text{Bias} \tag{5.2}$$

$$\text{Output} = \text{f(Z)} \tag{5.3}$$

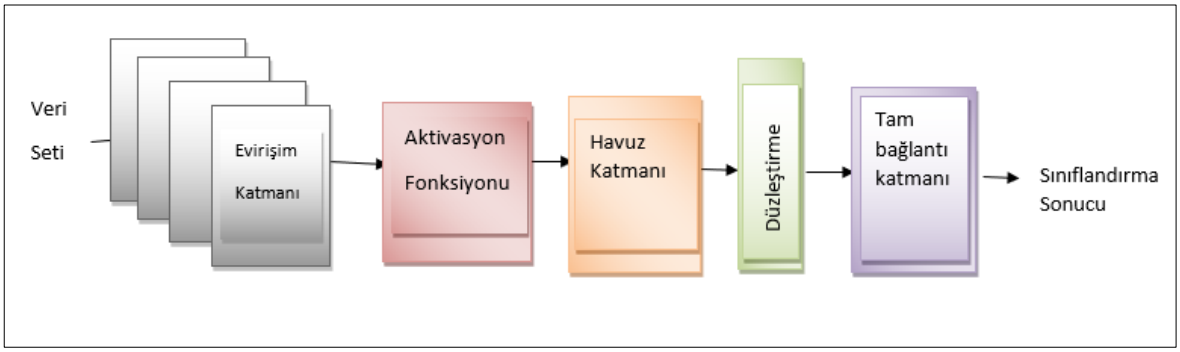
şeklindedir. Elde edilen bu Z değeri belirlenen aktivasyon fonksiyonunda işleme tabi tutularak çıktı üretilir. Sistem bu çıktıya göre kararını belirler. Literatürde en çok kullanılan aktivasyon fonksiyonları Sigmoid, Tanh, ReLU, Leaky ReLu, ELU şeklinde sıralanabilir. Aktivasyon fonksiyonlarının amacı girdilerin ağırlıklarını ve bias değerlerini ayarlamaktır. Aktivasyon fonksiyonlarını nöronların tetiklenmesini sağlar. Yüksek bir değer geldiği zaman fonksiyon yüksek bir değer üretecektir. Bunun sonucunda da nöron güçlü bir sinyal iletecektir. Öğrenmenin başarılı olup olmadığı anlamak için öğrenmenin ürettiği sonucun beklenen sonuca olan benzerliğine bakılır. Bu benzerliğin az olması durumunda öğrenmenin başarılı olduğu söylenemez. Bu benzerlik farkının hesaplanması için sinir ağlarında gider (cost) fonksiyonları kullanılmaktadır. Bu fonksiyonun ürettiği değerin büyüklüğü bir yönüyle öğrenmenin başarısızlığını gösterir. Öğrenmenin başarılı olması için fonksiyonunun ürettiği değerin sıfıra yaklaştırılması gerekmektedir. Bunun için bazı optimizasyonlar kullanılır. Kullanılan algoritmalara göre farklı gider (cost) fonksiyonu bulunmaktadır. Sonuç olarak bu fonksiyonlar sisteme hatasını gösterir. Sistemde bu hata değerine göre gerekli değişiklikler yapılarak sistemin istenilen sonuca ulaşması sağlanır. Derin öğrenme bilişim, sağlık, endüstri, askeri alanlar gibi çok farklı alanlardaki uygulamaları ve bu alanlardaki

problemlerin çözümünde başarılıdır. Derin öğrenme bilgisayar görüşü, model tanıma, görüntü ve ses işlem konularında başarıya ulaşmıştır. Ayrıca sınıflandırma ve problem tahminlerinde önemli gelişmeler elde etmiştir [60].

5.3. Derin öğrenme modelleri

Convolutional Neural Network (CNN)

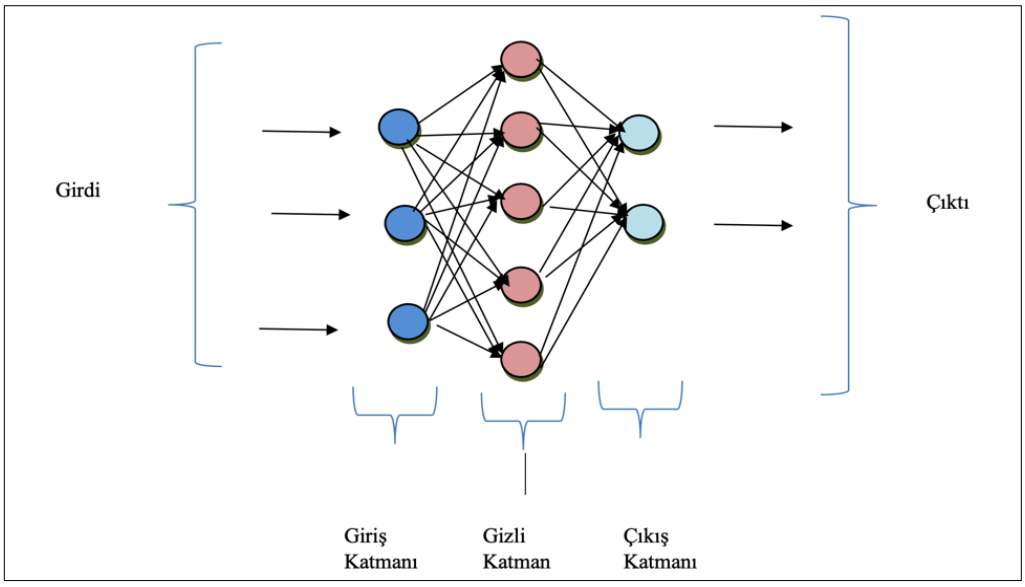
Convolutional Neural Network insanın görüş sisteminden esinlenerek geliştirilmiş bir sinir ağı modelidir. 2012 yılında yapılan ImageNet yarışmada CNN, Alexnet te kullanılması ile derin öğrenme ve CNN için yeni bir dönem başlamıştır. CNN çok katmalı olup her katmanda birden fazla evrişim ve havuz katmanları bulunmaktadır. Evrişimsel katmanlar girdilerden önemli özellikleri çıkararak girdiler arasındaki ilişkiyi ortaya koymaktadır. Bu ağ yapısı görseller ile ilgilendiğinden girdileri görsel piksellerden oluşur [63]. Havuz katmanları ile çıkarılan özellik haritasının boyutu azaltılır. Evrişimsel katmanın ve havuz katmanının sonuç ile tam bağlantılı bir ilişkisi vardır. Bu bağlantı sayesinde istenilen sonuç ortaya çıkarılması sağlanır. Bu sinir ağı görüntü ve pikselleri kullanarak bir öğrenme gerçekleşir. Bu yüzden günümüzde görüntü tanımlama, görüntü sınıflandırma, görüntü bölümlendirme, bütünlendirme ve görüntü çözümünün yeniden yapılandırması gibi işlemlerin yerine getirilmesinde başarılı ve popülerdir. Görüntü tespit ve diğer görüntü işlemlerinde kullanıldığı için eş ve gerçek zamanlı çalışması gerekmektedir. Sonuç olarak girdi görselini daha küçük parçalarla değerlendirip görseli farklı açılardan matrislerle temsil ederek benzerlikleri ve farklılıkları ortaya koyar [65]. Buna göre görseli tanıma ya da tanıma gerçekleştirilir. Şekil 5.3'te Evrişimsel sinir ağı algoritma modeli gösterilmiştir.



Şekil 5.3. Convolutional Neural Network algoritma şeması

Multilayer Perceptron (MLP)

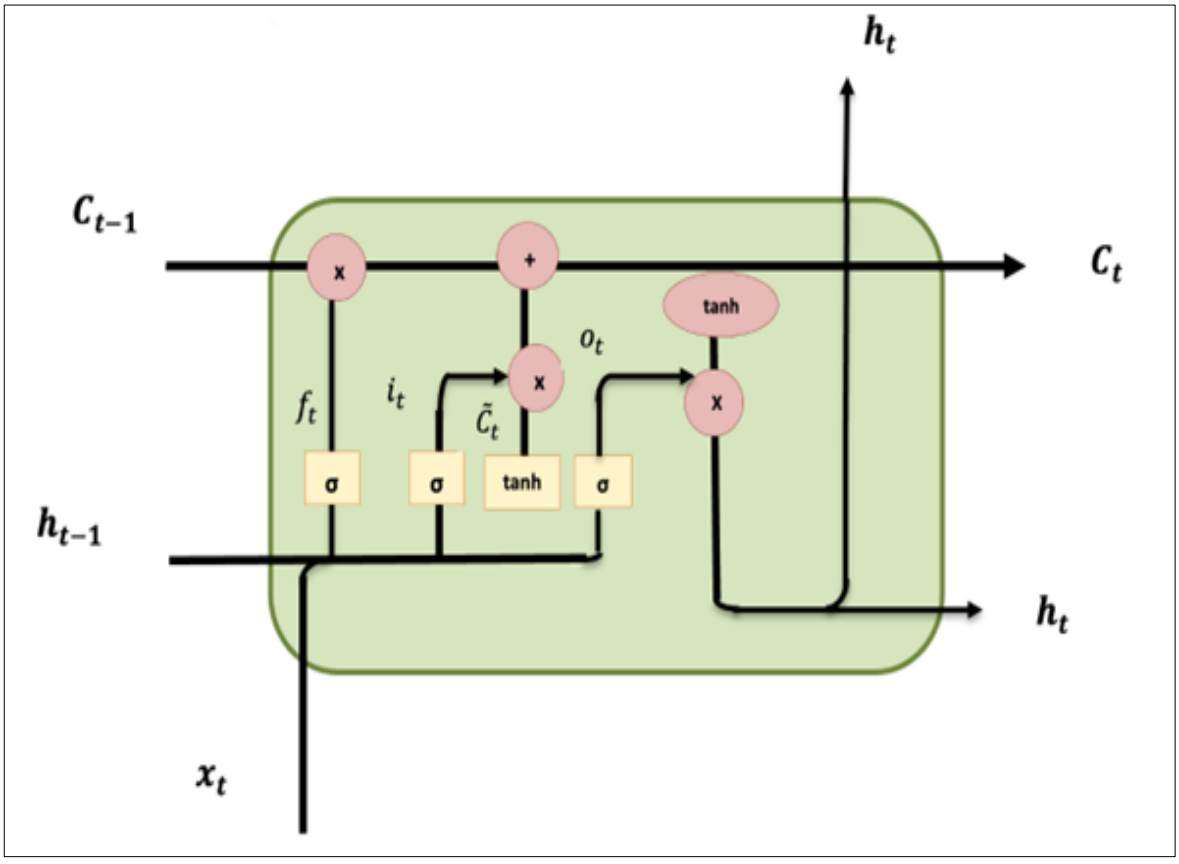
Multilayer perceptron (MLP) geri beslemeli yapay sinir ağlarının bir türüdür. Tek katmanlı algılayıcılar doğrusal sınıflandırmalar yapabiliyorken doğrusal olmayan daha karmaşık verileri sınıflandırmada problemler yaşamaktadır. Doğrusal olmayan verilerde tek katmanlı algılayıcılar başarısız olmaktadır. Bu noktada çok katmanlı algılayıcılar devreye girmektedir. Çok Katmanlı Algılayıcılar doğrusal olmayan sınıflandırmalarda ve XOR problemini çözmek için yapılan çalışmalar sonucu ortaya çıkmıştır. Çok Katmanlı Algılayıcı modeli Şekil 5.4'te gösterilmiştir. Çoklu girişlerde giriş katmanındaki tek nöron yeterli olmamaktadır. Paralel işlem yapan birden fazla nörona ihtiyaç duyulduğunda çoklu katman ile işlem yapmak gerekmektedir. Çoklu katmanlarda tek katmanlardan farklı olarak gizli katman yer almaktadır. Giriş katmanına gelen veriler giriş nöronlardan sonra ara katmanda bulunan nöronlara gönderilmektedir. Ara katman sayısı ve bu katmanlarda bulunana nöron sayısı ihtiyaca ve tercihe göre değişmektedir [62]. MLP modelinde en az bir ara katman olmak zorundadır. Giriş katmanı, gelen verileri katmanında bulunan nöronlar ile işleyerek çıktı olarak ara katmana göndermektedir. Ara katmanda aynı şekilde nöronlar ile veriyi işleyerek varsa sonraki ara katmana yoksa çıkış katmanına göndermektedir. Her katmanın çıkışı bir sonraki katmanın girişi olmaktadır. Böylelikle çıkışa ulaşılmaktadır. Çıkış katmanı sonucu üretmektedir. Üretilen sonuç birden fazla da olabilmektedir. Bu çıkış katmanında bulunan nöron sayısı ile ilişkilidir. Şekil 5.4'te MLP algoritma modeli gösterilmiştir.



Şekil 5.4. Multilayer Perceptron algoritma şeması

Long Short Term Memory (LSTM)

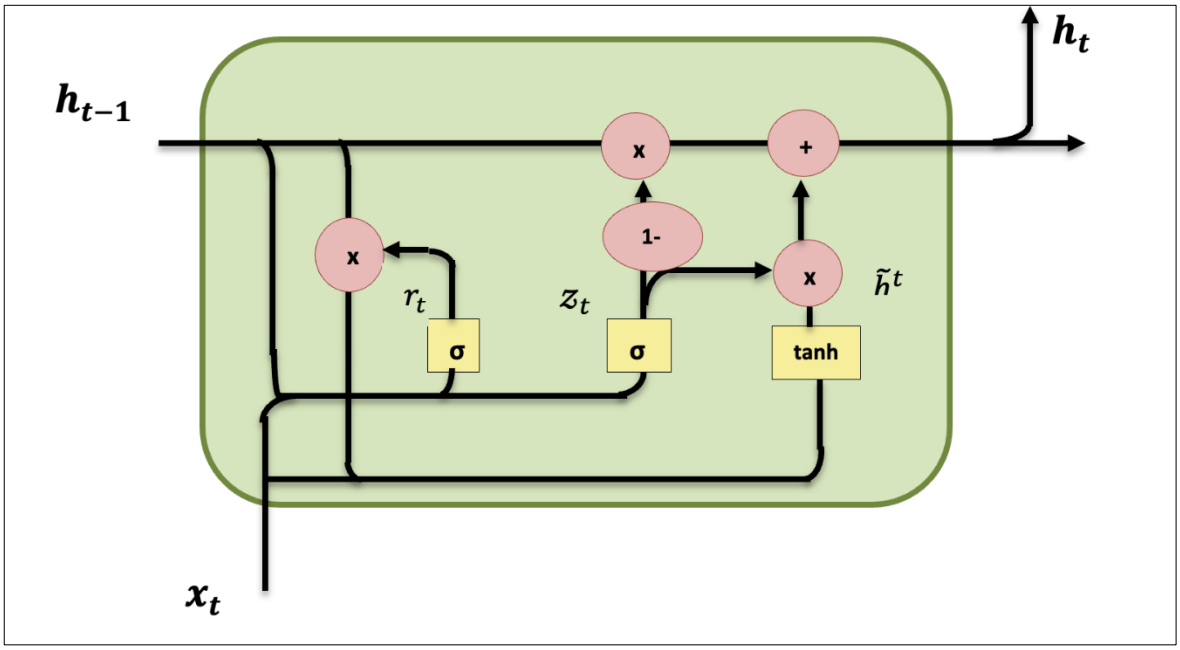
Yapay sinir ağı modeli olan LSTM bir RNN çeşididir. RNN modelinin özelleştirilmiş halidir. Temel RNN modellerinden farklı olarak geçmişe dönük uzun süreli geçmişe dönük verileri hafızasında tutarak karar mekanizmasında bu hafızadan yararlanmaktadır. RNN'e göre bu süre daha uzundur. LSTM, cell olarak adlandırılan hafıza bloklarından oluşmaktadır. Bu hafıza blokları LSTM'in temel bileşenidir. LSTM Forget Gate, Input Gate ve Output Gate olmak üzere üç kapıdan oluşmaktadır. İleri beslemeli sinir ağlarının aksine geri besleme ilişkisi bulunmaktadır. Veri seti elemanlarını ayrı ayrı ele almanın yanında veri setlerini beraber ele alıp işleyerek aralarında ilişki kurmaktadır. Yani zincirleme veri setlerinde kullanımı daha etkili olmaktadır. Bu zincirleme veri setine konuşma kayıtları, yazı türleri örnek gösterilebilmektedir [66]. LSTM algoritma modeli Şekil 5.5'te gösterilmiştir.



Şekil 5.5. LSTM algoritma şeması

Gated Recurrent Unit (GRU)

Model 2014 yılında kullanılmaya başlanmıştır. Temelinde basitleştirilmiş LSTM hücre mantığı yatmaktadır. Bu basitleştirme işlemi öğrenme performansını arttırmaktadır. Bu avantajlı durum çoğu geliştirici tarafından bu modelin kullanılmasını sağlamıştır. GRU LSTM gibi bilgi akışını kontrol etmek için kapıları(gate) kullanmaktadır. GRU LSTM'nin bir varyasyonu olarak da düşünülebilir, çünkü her ikisi de benzer şekilde tasarlanmıştır. GRU konuşma tanımada, bellek ve kümeleme ile ilişkili makine öğrenme görevlerini gerçekleştirmek için bağlantıları bir dizi düğüm aracılığıyla tespit edip kullanmayı amaçlayan sinir ağı modelinin parçasıdır. Yukarda belirtilen basitleştirme işleminde öncelikle iki durum vektörü tek bir vektör olarak birleştirilir. Geçit denetleyicisi hem hafızadan silmeyi hem de giriş kapısını kontrol etmektedir. Denetleyici 1 çıktısını verirse giriş kapısı açıktır. O çıktısını verirse giriş kapısı kapalıdır. Giriş kapısı açık iken silme kapısı kapalıdır. LSTM mantığının basitleştirilmiş modeli olmasından dolayı konuşma tanımada, insan genomunda, el yazısı analizi gibi uygulamalarda kullanılabilmektedir [67]. GRU algoritma modeli Şekil 5.6'da gösterilmiştir.



Şekil 5.6. GRU algoritma şeması

5.3.2. Derin öğrenme fonksiyonları

Derin öğrenmede en sık kullanılan fonksiyonlar aşağıdaki şekilde sıralanmıştır.

Sigmoid fonksiyonu

Sigmoid fonksiyon gelen değeri 1 ve 0 arasında bir değerle çıktı olarak gönderir yani gelen değeri bu aralık temsil eder. En yüksek değer geldiğinde 1 ile temsil edilirken en düşük değer ise 0 ile temsil edilmektedir. Çıktı değeri 0 altına inmemektedir [64].

Tanh

Tanh fonksiyonlarında ise gelen değerler 1 ile -1 aralığında bir çıktı ile gösterilir. LSTM ve GRU gibi bazı sinir ağlarında yaygın olarak kullanılmaktadır.

ReLu (Rectified Linear Unit)

ReLu ilkez 2012 yılında Alexnet ile kullanılmaya başlanmıştır. Günümüzde literatürde sıkça kullanılan bir fonksiyondur. Fonksiyona gelen değerin negatif ve pozitif durumlarına bakarak sonuç üretmektedir. Eğer gelen değer negatif ise fonksiyon 0 değerini üretmektedir. Gelen değer pozitif ise herhangi bir sıkıştırma ve değiştirme işlemi uygulamadan doğrudan değeri sonuç olarak göstermektedir. İşlem basitliği sayesinde yüksek donanım ihtiyacı duymamaktadır.

Leaky ReLu

ReLu fonksiyonun değiştirilerek geliştirilmiş bir dağıtımıdır. ReLu fonksiyonlarında negatif gelen girdiler doğrudan 0 çıktısı vermesinden dolayı bazı nöronlar işlevsiz hale gelmektedir. Leaky ReLu fonksiyonlarında bunun önüne geçmek için negatif girdiler belirli oranda küçültülerek çıktı olarak verilmektedir. Negatif girdileri büyük oranda küçültür ve farklı negatif çıktı değerleri verir.

ELU (Exponential Linear Unit)

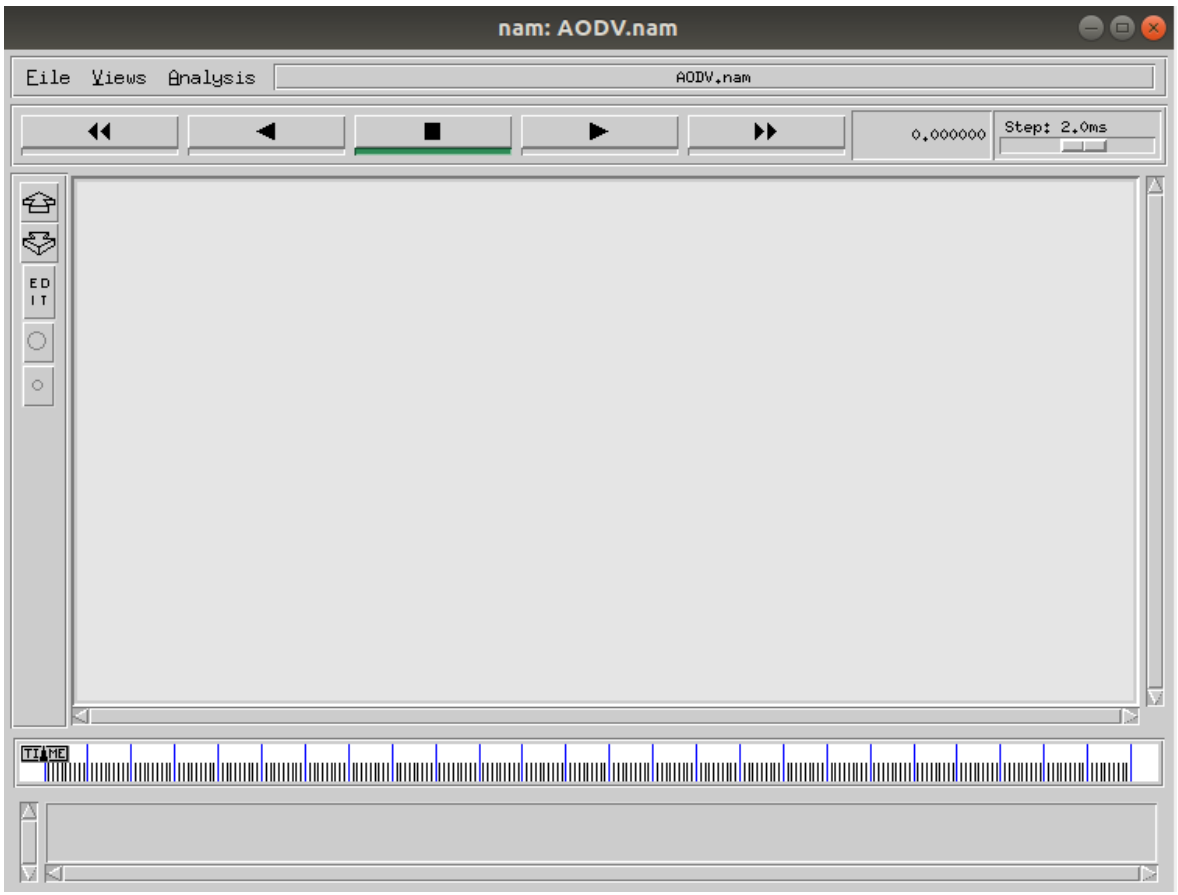
ReLu fonksiyonun tüm avantajlarının sağlamaktadır. Negatif değerler 0 çıktı değeri almadığından işlevsiz nöronlarının önüne geçilir. Sıfırdan küçük girdi değerleri exponential

bir fonksiyon kullanılarak çıktı değeri üretilmektedir. Bu işlemden dolayı ReLu fonksiyonundan daha yavaştır.

6. ÇALIŞMADA KULLANILAN ARAÇLAR

6.1. NS-2

Sensör ağları, gerçek ortamda çalıştırılmadan önce olası sonuçları görebilmek için farklı benzetim ortamlarında denenmektedir. Literatürde NS-2, OMNeT++, OPNet, CNet, SensorSim, Tossim, ATEMU, CupCarbon gibi benzetim ortamları kullanılmaktadır. Bu çalışmada NS-2.35 (Network Senario) benzetim ortamı kullanılmıştır. 1989 yılında NS-2 Real Network Simülation'ın bir dağılımı olarak kullanılmaya başlanmıştır. NS-2.35 platformunun çeşitli versiyonları bulunmaktadır. NS-2 ortamı ile Tool Command Language (tcl), network animasyon (nam) ve trace (tr) dosyaları beraber kullanılmaktadır. Senaryo C++ dili kullanılarak .tcl uzantılı script dosyasına kaydedilerek çalıştırılmaktadır. Dosyanın uzantısı tcl olsa da dosya içerisinde kullanılan dil C tabanlıdır. Şekil 6.1'de NS-2 arayüzü gösterilmiştir.



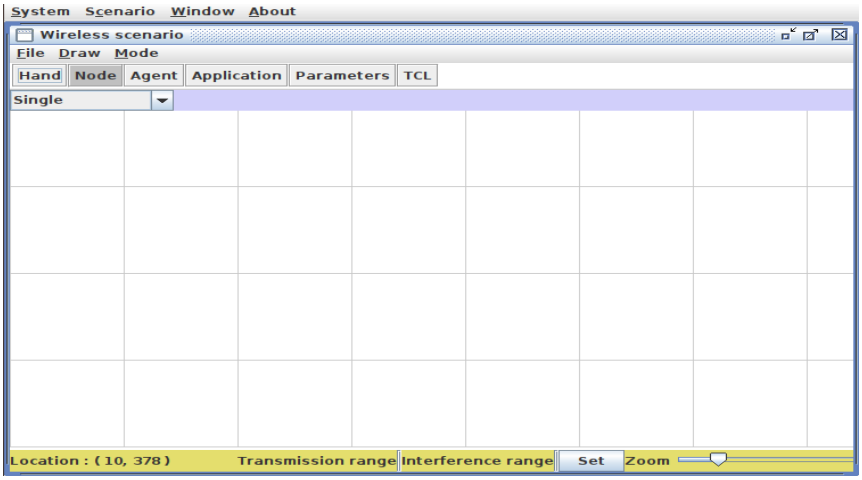
Şekil 6.1. NS-2 benzetim ortamı arayüzü

Tcl web ve masaüstü uygulamalarında, ağ programlarında ve gömülü uygulamalarda kullanılmaktadır. John Ousterhout tarafından Berkeley üniversitesinde geliştirilmiştir. Tcl script dosyasının çalıştırılmasının ardından .nam ve .tr (trace) uzantılı dosyalar otomatik olarak üretilmektedir. Bu dosyalardan .nam dosyası düğümlerin simülasyon ara yüzünü çalıştırmak ve görüntülemek için üretilmektedir. Nam dosyası açılmadan simülasyon ara yüzü görüntülenememektedir.

Benzetim ortamı çalıştırıldığında .nam dosyası ile paralel trace (.tr) dosyası üretilmektedir. Ancak trace dosyasının içeriği uygulamanın tamamlanmasının ardından tamamlanmaktadır. Ağ çalışırken herhangi bir hata alınması ya da programın kapatılması durumunda trace dosyası o ana kadar gerçekleştirilen trafik bilgilerini kaydetmektedir. Trace dosyası, ağ trafiği, ağ paketleri ve düğümler hakkında detaylı bilgiler içermektedir. NS-2 platformunun tüm versiyonlarında ağ trafiğini incelemek isteyen geliştirici trace dosyası ile işlem yapmaktadır. Kullanılan düğüm sayısı ağ trafik yoğunluğu, ağın çalışma süresi, trafik türü gibi değişkenlere bağlı olarak dosya boyutu değişiklik göstermektedir. Dosyada bulunan satırların boyutu içerdiği bilgilere göre farklı uzunlukta olabilmektedir.

6.2. Ağ Senaryo Üreticisi

NSG arayüzü: ağ mimarisi, iletişim protokolü, paket boyutu, düğüm sayısı, düğüm konumu, düğüm hareketi, ağın çalışma süresi gibi özelliklerin tasarlandığı java tabanlı bir araçtır. NSG ara yüzü Şekil 6.2'de gösterilmiştir.

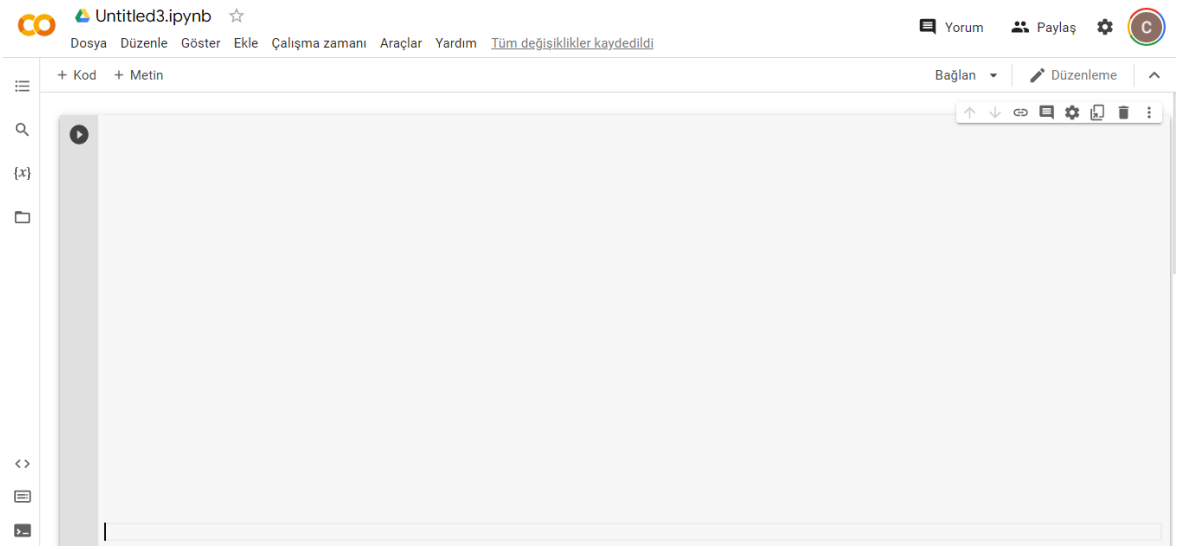


Şekil 6.2. NSG aracı arayüzü

NSG2.1 ara yüzünde öncelikle senaryo seçimi yapılması gerekmektedir. Scenario butonu ile kablolu ya da kablosuz senaryolar geliştirilebilmektedir. Node butonu ile düğümlerin nasıl bir düzen ve hangi sayıda konumlandırılacağı belirlenmektedir. Kullanıcı isterse tek tek kendi istediği şekilde düğümleri ortama ekleyebildiği gibi, isterse yatay ya da dikey seçeneği tıklayarak bir düzen halinde ekleyebilmektedir. Agent butonu ile iletim tipi belirlenmektedir. Application butonu ile simülasyonun başlama bitiş süreleri girilmekte ve uygulama tipi seçilmektedir. Parametre butonu ile uygulamanın protokolü ve iletim değerleri girilmektedir. Bu bölümde protokol tipi (AODV, DSR, TORA, DSDV), kablosuz özellikleri değiştirme ve kanal seçenekleri bulunmaktadır. Ara yüzde yapılan işlemlerin kaydedilmesi için önce Save As sonra da Done butonu tıklanmaktadır. Tcl butonu ile oluşturulan trafik bir dosya olarak kaydedilip çalıştırılmaya hazır hale getirilmektedir.

6.3. Google Collabratory

Makine öğrenme modelleri ya da derin öğrenme modelleri kullanılarak yapılan veri setinin analizinde geliştirici iki yol ile veri analizini gerçekleştirebilmektedir. İlk yol kullanıcının kendi bilgisayarında gerekli kurulumları yaparak analiz işlemini yerelde gerçekleştirmesidir. Bunun için Anaconda, Visiula Code Studio, Eclipse gibi platformların kurulumunu yaparak veri setini kendini bilgisayarında analiz edebilmektedir. İkinci yolda ise Google Collabratory Microsoft Azure gibi bulut servislerinden yararlanarak işlemlerini servis sağlayıcılarının sunucularında gerçekleştirebilmektedir. Bu yöntemde kuruluma gerek yoktur. Azure platformunda CPU üzerinde işlemler yapılabilirken Collabratory'de GPU/CPU üzerinden işlem yapılabilmektedir. Bu çalışmada Google Collabratory tercih edilmiştir. Google Collabratory yazılım alanında çalışan geliştiriciler için Google tarafından sunulan bir bulut hizmetidir. Özellikle kullanıcıların sınırlı donanımsal kaynağının yetersiz kaldığı durumlarda ya da yüksek performanslı donanıma ihtiyaç duyan programların çalıştırılmasında kullanıcıya büyük kolaylık sağlamaktadır. Özellikle son yıllarda derin öğrenme gibi veriyi analiz ederken yüksek performanslı donanıma ihtiyaç duyan programlar için sıkça kullanılmaktadır. Şekil 6.3'te Google Collabratory arayüzü görülmektedir.



Şekil 6.3. Google Collabratory arayüzü

Google'un diğer uygulamaları gibi Google Drive'ın içinde yer alan ve Drive ile senkron çalışan bir platformdur. Servisi kullanabilmek için Gmail kullanıcı hesabı olması gerekmektedir. Platformu kullanmadan önce mail adresi üzerinden çift doğrulama ile kimlik doğrulaması yapılmaktadır. Platform üzerinde Python 2/3 ve R dilleri ile çalışma yapılabilmektedir. Şekil 6.3'de Ipython notebook arayüzü açıldıktan sonra ayrı ayrı hücrelere metin ya da kod yazılabilmektedir. Veri seti analiz edilecek ise veri setinin Google Drive'da yüklü olması gerekmektedir. Kullanıcının tercihine göre program CPU-GPU üzerinden çalıştırılabilmektedir. Platform üzerinde özellikle bu çalışmada olduğu gibi Keras, TensorFlow, PyTorch ve OpenCV gibi kütüphaneleri kullanarak derin öğrenme uygulamaları geliştirebilmektedir. Platform çevrimiçi çalıştığı için bağlantı kopması durumunda hata mesajı alınmaktadır. Ayrıca her bağlantı koptuğunda ya da platform açılıp kapatıldığında çift doğrulama ile kimlik doğrulaması yapılmaktadır. Bağlantının kopup kopmadığı arayüzün sağ üst köşesindeki bağlan butonunun aktif olması ile anlaşılabilmektedir. Google sunucuları ile bağlantı sağlandıktan sonra gerekli kütüphaneler yüklenerek analiz işlemleri gerçekleştirilebilmektedir.

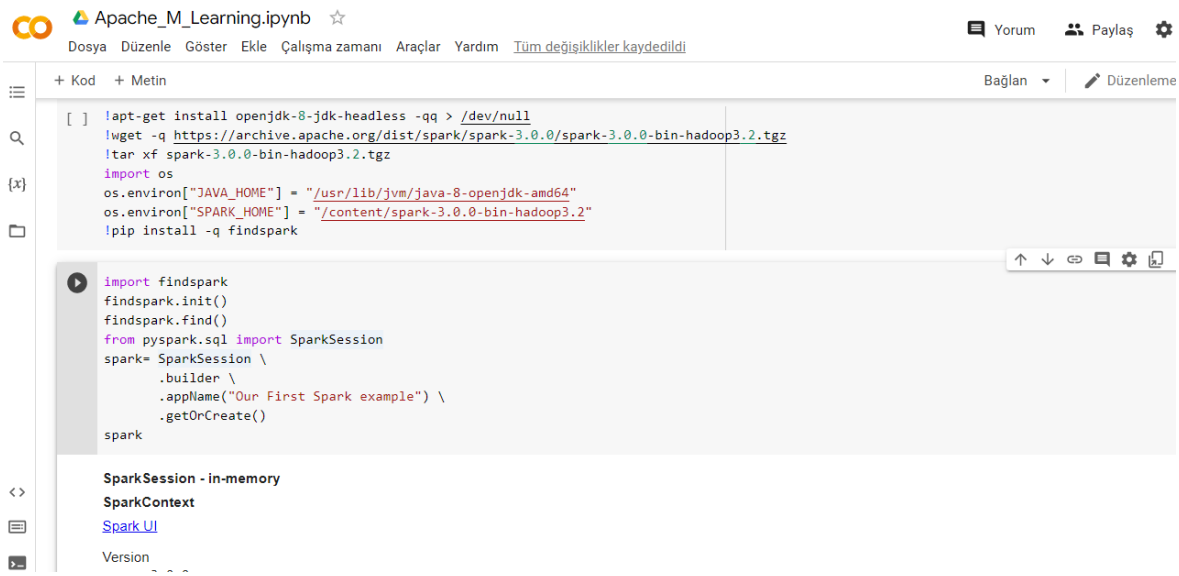
6.4. Apache Spark

Apache Spark tekli ve çoklu paralel çalışabilen, veri işleme mühendisliğinde, veri biliminde, makine öğrenmelerinde kullanılan çoklu dil motorudur. Java, Python, Scala, R ve Sql dilleriyle çalışabilmektedir. Büyük veriler dağıtık olarak analiz edilebilmektedir. Herhangi

bir depolama birimi bulunmamaktadır. Analizlerini Ram üzerinde gerçekleştirmektedir. Apachi Spark alternatifi olan Hadoop kütüphanesi göre 110 kat daha hızlı çalışmaktadır [68]. Apache Spark'ın önemli özellikleri ilgili uygulamanın işlem hızını arttıran, hafıza içinde küme hesabı yapan ve tüm kümeleri programlamak için örtük veri paralelliği ve hata toleransı sağlayan bir çerçeve olmasıdır. Apache Spark farklı ve paralel iş uygulamalarında, tekrarlı algoritmalarda, etkileşimli sorgularda çalıştırılarak sonuç üretmek üzerine dizayn edilmiştir. Apache Spark özellikle büyük verilerin işlenmesinde diğer araçlara göre daha hızlıdır. Bu özelliği sayesinde birçok geliştirici tarafında tercih edilmektedir. Diğer araçlara göre hızlı olmasının nedeni paralel işleme ve veriyi kontrollü bölümleme yoluyla işlemesidir. Programlama katmanının basit olması güçlü bir ön bellek sağlamaktadır. Apache Spark'ın gerçek zamanlı çalışması en önemli avantajlarındandır. Bunu belek içi hesaplama özelliği sayesinde yapmaktadır. Spark ekosistemi çeşitli alt bileşenlerden oluşmaktadır. Bu bileşenlerin ayrı ayrı sağladığı avantajları Spark bir bütün olarak geliştiricilere sunmaktadır. SparkCore, büyük verilerin işlenmesi için tasarlanmış paralel ve dağıtık çalışan veri işleme motorudur. Spark'ın çekirdeği olmasından dolayı bellek yönetimi, hata ayıklama, kümedeki işleri planlama, yönetme, takip etme gibi yönetim işlerinden sorumludur. Spark Streaming, gerçek zamanlı gelen veriyi hızlı bir şekilde işleyerek sistemin gerçek zamanlı çalışılmasını sağlamaktadır. Bunu yaparken yüksek verim ile sonuçlar üretmektedir. Spark GraphX, verilerin görselleştirilmesini sağlayan bir bileşendir. Kafka, HDFS, Hbase farklı yerlerdeki verinin işlenerek görselleştirilmesini sağlamaktadır. RDD bileşenlerini kullanarak küme üzerinde verileri hesaplanmasını sağlamaktadır [68].
Spark R, büyük verinin işlemesi gerçekleşirken verinin seçimi, filtrelemesi, toplanması gibi işlemlere destek sağlayan R paketleri olarak çalışmaktadır.

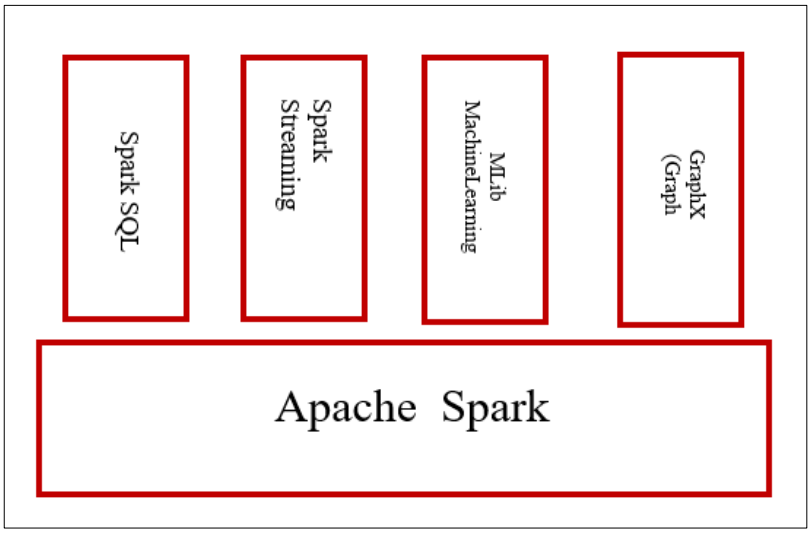
MLib, Apache de bulunan bu kütüphane ile geliştiriciler makine öğrenme modellerini, yardımcı araçlarını kullanabilmektedir [68]. Kütüphane import işlemin ardında tüm özellikleri kullanılabilmektedir.

Farklı bulut platformlar üzerinde Apache Spark'ın ve özelliklerinin kullanılabilmesi için oturum kurulması gerekmektedir. Şekil 6.4'te Google Collabratory de Apache Spark ile gerçekleştirilen çalışmaya ait arayüz yer almaktadır.



Şekil 6.4. Apache Spark kullanım kodları

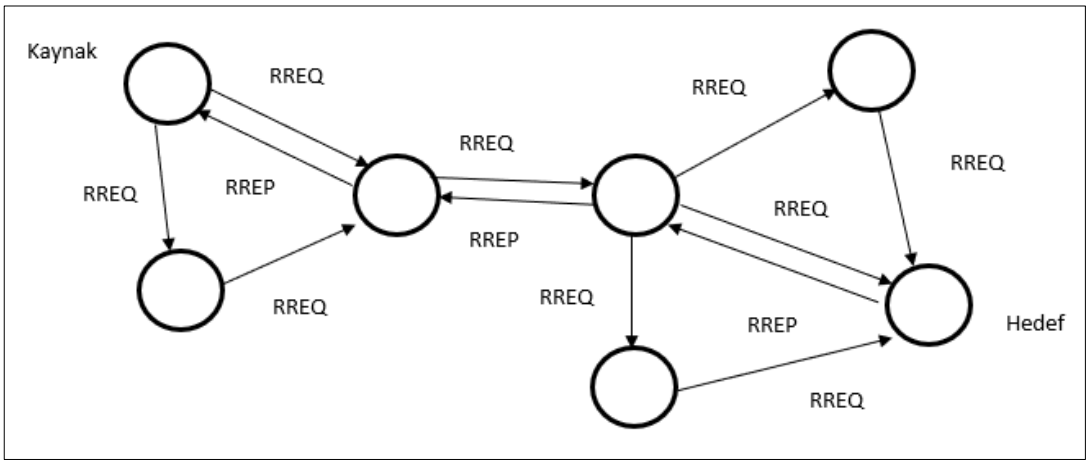
Çalışma, Google Colab ortamında Apache Spark büyük veri platformu üzerinde Pyspark aracı kullanılarak yapılmıştır. Makine öğrenmesi ve derin öğrenme algoritmaları için PySpark MLib kütüphanesi, Scikit-learn ve Keras kütüphaneleri kullanılmıştır. Şekil 6.5'de Apache Spark mimarisine yer verilmiştir.



Şekil 6.5. Apache Spark mimarisi

7. GELİŞTİRİLEN UYGULAMA

KSA'da katmanlara göre güvenlik zafiyetleri olduğu gibi her geçen zamanda yeni saldırı modelleri ve konseptleri geliştirilmektedir. Sensör ağ geliştiricileri bu saldırılar için yeni tespit ve çözümler geliştirmekte ve literatüre yeni çalışmalar kazandırmaktadır. Saldırı tespiti ve çözümleri için saldırı trafik veri seti büyük önem arz etmektedir. Elde edilen veri seti geliştiriciler tarafından çeşitli model, yaklaşım ve algoritmalarla kullanılarak saldırı tespit yapmaktadır. Çalışmada öncelikle Windows işletim sistemi üzerine Virtual Box platformu kurulmuştur. Virtual Box platformunun üzerine de Ubuntu 18.04 sanal makinesinin kurulumu yapılmıştır. Sanal makine özellikleri tanımlanırken benzetim ortamında çalıştırılacak senaryonun kompleks olması göz önünde bulundurularak sanal makineye 4 GB ram ve 100 GB bellek tanımlanmıştır. Ubuntu işletim sistemi üzerine NS-2 benzetim ortamı ilgili kodlar ile yüklenmiştir. Komut girme işlemi Ubuntu terminal üzerinde yönetici rolü ile gerçekleştirilmiştir. NS-2 yükleme işlemi tamamladıktan sonra yüklemenin başarılı olup olmadığını tespit etmek için sağlama komutları kullanılmıştır. NS-2 platformunun kurulumundan sonra senaryoda bulunan düğümlerin saldırgan düğümlere dönüşmesi için kullanılan 'malicious' tanımlaması aodv.cc ve aodv.h protokol sistem dosyalarında tanıtılmıştır. Sistem dosyalarında belirtilen değişiklikler ve eklemeler yapıldıktan sonra .tcl dosyasında belirlenen düğümlere belirlenen zamanda malicious tanımlaması yapılarak bu düğümler üzerinden Blackhole ve Selective Forwarding saldırıları gerçekleştirilmiştir. Yukarıda belirtilen sistem dosyalarında değişiklik yapılmasının nedeni senaryoda AODV protokolünün kullanılmış olmasıdır. AODV protokolünün çalışma modeline Şekil 7.1'de yer verilmiştir.



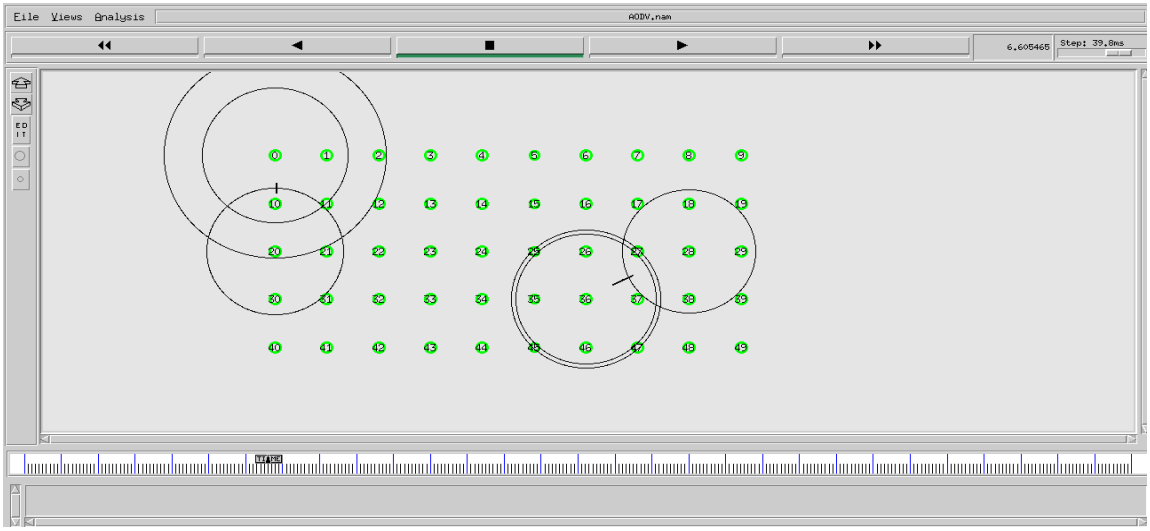
Şekil 7.1. AODV çalışma mimarisi

Şekil 7.1'de görüldüğü gibi iletişim ara düğümler üzerinden uçtan uca gerçekleşmektedir. Senaryoda düz ağ mimarisi ve uçtan uca hiyerarşik olmayan bir ağ yapısı planlandığı için AODV protokolü tercih edilmiştir. Düğümler yön oluşturmak için, hangi düğümün nerde olduğunu öğrenmek için ve hedefe hangi alternatif yollardan gidebileceğini bulmak için Şekil 7.1'de olduğu gibi RREQ paketleri göndermektedir. AODV protokolü NS-2 benzetim ortamında DSDV ve DSR protokolleri gibi yüklü olarak gelmektedir. Senaryo oluşturulurken MANET (Mobile Ad Hoc Networks) protokollerinden olan AODV (Adhoc On Demand Vector) protokolü ile ağ trafik yönlendirmesi gerçekleştirilmiştir. MANET (Mobile Ad Hoc Networks) kendi kendine senkronize olabilen kablosuz ağlarda kullanılan, düğümler arası iletişimi sağlayan ve sabit bir alt yapı ve omurgaya ihtiyaç duymayan teknolojidir [69]. AODV protokolü diğer MANET protokollerinden farklı olarak periyodik yol kontrol mesajları göndermektedir [70]. Bu protokolde her düğümün hafızasında belirli bir süre saklanmak üzere yön tablolarını bulunmaktadır [71]. Düğümler arasında iletişim; belirlenen bir yol üzerinden gerçekleştirilir. Bu yol protokolün algoritması tarafından belirlenir. Algoritma uzaklık, iletim hızı, geçilecek düğüm sayısı gibi belirli parametrelere bakarak yolu belirlemektedir. AODV protokolü hiyerarşik ağ yapısında tercih edilmemektedir. Hiyerarşik protokollerin aksine AODV protokolünde iletişim sağlanırken öncelikle küme başı düğüm ile değil ara düğümler yardımıyla doğrudan verinin iletileceği hedef düğüm ile uçtan uca gerçekleştirilir.

AODV protokolü reaktif ve istek üzerine (on demand) çalışan protokolüdür [70]. Protokol Belmann Ford modelini temel almaktadır [69]. Reaktif protokol olduğundan ihtiyaç olması durumunda yönlendirme tablosu oluşturur [9]. Bu tabloyu oluştururken hangi düğüm kaç adım ötede, hedefe giden alternatif yollar hangileri gibi bilgileri toplamak için yön istek paketleri (RREQ) göndermektedir [70]. Yön istek paketinde; kaynak düğümün adresi, hedef düğümün adresi, geçilen düğüm sayısı gibi bilgiler bulunmaktadır. Bu paket kaynak düğümden tüm düğümlere gönderilmektedir. Paket sayesinde toplanan bilgiler kaynak düğümün yön tablosuna kaydedilerek sonraki iletişimde kullanılmak üzere belirli bir süre saklanmaktadır. Tablodaki bu bilgiler düğümlerin sınırlı hafızaları için fazladan bir yük oluşturmaktadır. Bu durum AODV protokollerinin olumsuz bir yanıdır. Kaynak düğümün kaydettiği bu bilgilerden hedef düğüm için yol tespiti yapılırken istek paketlerinden sıra numarası en güncel ve hedefe ulaşana kadar geçilecek düğüm sayısı en az olan tercih etmektedir [70]. Yön oluşturulduktan sonra belirli periyotlarla iletişimin devamlılığı için yol kontrol edilmektedir. İletişim için yön ve güzergâh belirlendikten sonra iletişim başlamadan

son doğrulama denebilecek merhaba paketi gönderilerek doğruluğu onaylandıktan sonra asıl iletişim başlamaktadır. Oluşturulan iletişim yolunda kopmalar olursa hata mesajı yayınlanarak yön işlemleri tekrar oluşturulur. İstek paketleri yönlendirme için önemlidir. Ancak ağdaki tüm düğümlere gönderilmesinden dolayı ağ trafiğine fazladan yük getirmektedir. Özellikle amacı dışında kullanılması durumunda sensör ağlarının çalışmasını aksatır ya da engeller. İstek paketlerinin manipüle edilmesi; ağda yoğunluk oluşturularak trafiğin gecikmesine, fazla miktarda trafik oluşmasından dolayı çakışmalara-paket düşürmelerine, yoğun trafikten dolayı düğümlerin enerjilerinin kısa sürede tüketmesine sebep olmaktadır.

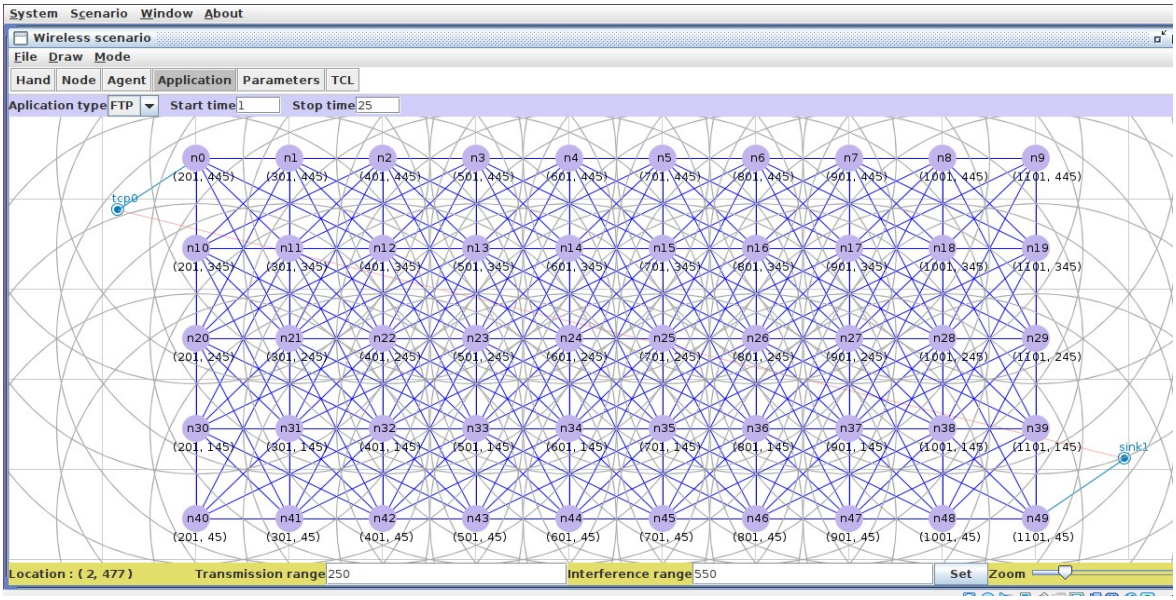
Senaryonun çalışacağı NS-2 platformu kurulumu tamamlandıktan sonra senaryonun hazırlama aşamasına geçilmiştir. Senaryo Blackhole, Flooding, Selective Forwarding ve Normal olmak üzere toplamda 4 farklı ağ trafiği olacak şekilde tasarlanmıştır. Senaryo temelde NSG 2.1 aracı ile tasarlanmasına rağmen sonraki aşamalarda ağ ve düğüm ile ilgili değişikliklerin tamamı .tcl dosya üzerinden gerçekleştirilmiştir. KSA ağ senaryosu oluşturulurken 42 denemenin ardından ideal istenen şekle ulaştırılmıştır. Şekil 7.2'de NS-2 ara yüzü görülmektedir.



Şekil 7.2. NS-2 arayüzü

Yeşil renkli daire ve sayılar düğümleri temsil etmektedir. Düğümler etrafında oluşan daireler ise anlık paket gönderimini göstermektedir. Senaryo planlanırken Java tabanlı NSG 2.1 aracı kullanılarak ağ ve düğümlere ait temel tüm özellikleri tanımlanmıştır. Ara yüz sayesinde ağa ait; ağ topoloji şekli, ağ çalışma süresi, ağda kullanılacak olan protokol, bu protokolde

kullanılacak paket boyutu, birim zamanda gönderilecek paket miktarı ve düğümlere ait; düğüm sayısı, düğüm hareket durumu, düğüm enerji miktarı, düğümlerin konumlandırılması gibi bilgiler girilerek .tcl uzantılı olarak kaydedilmiştir. Şekil 7.3'de NSG 2.1 aracının ara yüzü gösterilmiştir.



Şekil 7.3. Network Senario Generator aracı arayüzü

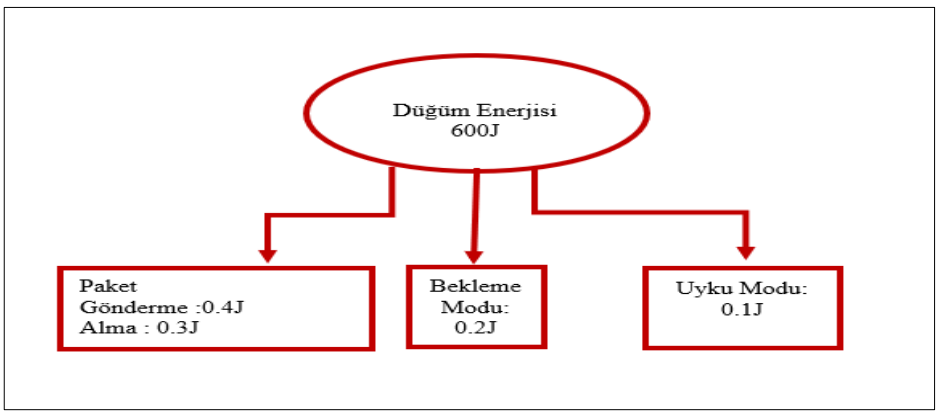
Senaryoda 200 düğüm 1600 dakika boyunca çalıştırılarak ağ veri trafiği elde edilmiştir. Düğümler 2000 x 1000 dikdörtgen bir topolojik alana konumlandırılmıştır. Düğümler parametrelere göre değerlendirilirken objektif olunması için düğümlere aynı parametrelere aynı değerler tanımlanmıştır. Senaryoda düğümlerin hareket etmesi zararlı düğümleri saldırıyı başarılı şekilde gerçekleştirmesine engel teşkil etmektedir. Bu yüzden düğümler hareketsiz olarak konumlandırılmıştır Ağ ve düğümlere ait özelliklere Çizelge 7.1'de yer verilmiştir.

Çizelge 7.1. Senaryo parametreleri

Ağ Parametreleri	Aldığı Değerler
Düğüm sayısı	200
Time	1600 dakika
Konumlandırma Alanı	2000x1000
Paket Boyutu	512
Paket Başlığı	25
Kullanılan Protokol	AODV
Maximum Transfer Uzaklığı	500m
Mac Protokolü	CSMA
Başlangıç Enerjisi	600J
Zararlı Trafik Yoğunluğu	% 16

Çizelge 7.1'de görüldüğü gibi 200 düğüm 1600 dakika 2000 x 1000 metre alana konumlandırılarak NS-2 ortamında AODV protokolü kullanılarak çalıştırılmıştır. Düğümlerin çalışma süresi ile başlangıç enerji miktarı arasında doğrudan ilişki bulunmaktadır. Senaryo çalışma süresi arttıkça enerji kullanımı artmaktadır. Ayrıca enerji değeri, zararlı düğümler ile trafiğin üreten-ileten düğümler göz önüne alınarak belirlenmiştir. Diğer taraftan veri setinin istenen boyutta ulaşması için çalışma süreside önem arz etmektedir. Bu yüzden düğümlerin enerji değeri belirlenirken veri seti göz önünde tutularak spesifik olarak belirlenmiştir.

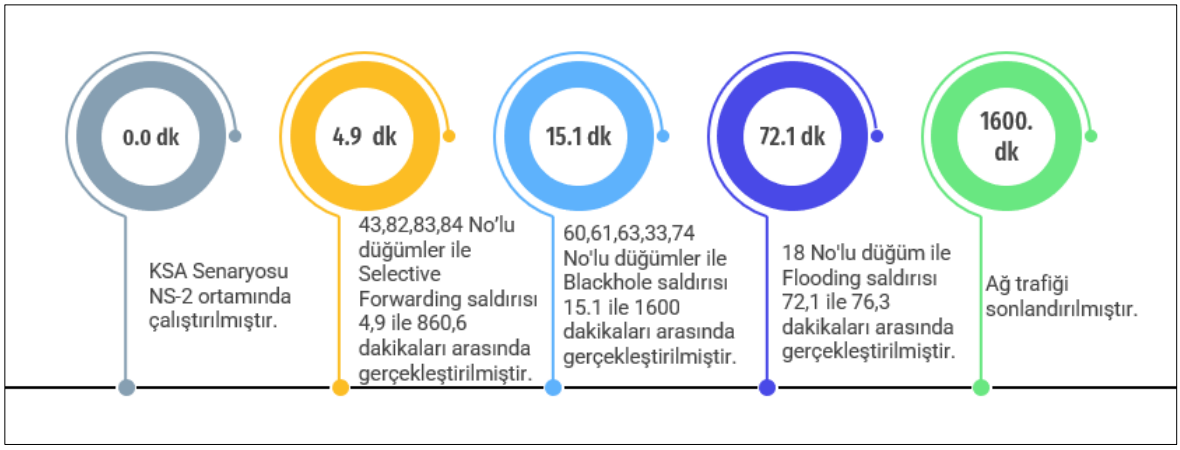
Düğümlerin ilk enerji değerleri 600 J olarak belirlenmiştir. Bu değer belirlenirken çalışma defalarca tekrarlanarak gerekli enerji değeri tespit edilmiştir. Düğümlerin enerji tüketim miktarlarının bulundukları durumlara göre değişimi Şekil 7.4'te gösterilmiştir.



Şekil 7.4. Düğümlerin durumlara göre enerji kullanım miktarı

Şekil 7.4'te görüldüğü gibi düğümlerin başlangıç enerjisi 600 joule olarak belirlenmiştir. Düğümler paket gönderirken 0,4 joule, paket alırken 0,3 joule, ağda hazır durumda beklerken 0,2 joule, uyku durumunda geçtiklerinde ise 0,1 joule enerji harcamaktadırlar. Enerjide gerekli ayarlamalar gerçekleştirildikten sonra ağ ve iletişim ile ilgili işlemlere geçilmiştir. Uygulamada normal ağ trafiği dahil 4 farklı ağ trafiği gerçekleştiği için herhangi trafik dengesizliği diğer trafiklerinde çalışmasını etkilemekte, program hata vermektedir. Bu yüzden her trafikte gönderilen paket boyutu ve paket gönderim zaman aralığı gibi değerler senaryonun planlanan şekilde tamamlanması için özel olarak belirlenerek .tcl dosyasına kaydedilmiştir. Ayrıca paket iletim sıklığı dengeli olarak ayarlanmıştır. 200 düğümlü bir ağ oluşturulduğu için benzetim ortamında çalıştırılırken yukarıda belirtilen parametrelere uygun değerler verilmediği takdirde sıklıkla hata mesajı alınmıştır. Bu yüzden .tcl dosyasında gerekli düzeltmeler yapılarak en uygun değerler ile senaryo dosyası hazır hale getirilmiştir. Senaryoda ilk olarak Selective Forwarding saldırısı gerçekleştirilmiştir. Selective Forwarding saldırısı veri bütünlüğüne yönelik yapılan bir saldırıdır. Forwarding saldırısı DoS saldırı çeşidine örnektir. Forwarding saldırısı için 0 no lu düğüm ile 121 no lu düğümler arasında trafik oluşturulmuştur. 0 no lu düğüm (Kaynak) ile 121 no lu düğüm (Hedef) arasındaki 43,82,83,84 no lu saldırgan düğümler trafiği manipüle ederek kendisine iletilmesi için gelen paketlerin bir kısmını iletirken bir kısmını düşürmektedir. Selective Forwarding atağı 4,9 dakika başlayarak gerçekleştirilmiştir. Uygulamada atağı gerçekleştirirken her paketin düşürülmemesi ya da her paketin gönderilmemesi için paket düşürme işlemi her 30-40-80 saniye periyotlar ile yapılması kodlanmıştır. İlgili saldırgan düğümlere belirli sürelerde normal düğüm gibi davranmaları belirli sürelerde ise paket düşüren zararlı düğüm gibi davranmaları kodlanmıştır. Çalışmada gerçekleştirilen ikinci saldırı olan Blackhole saldırı türü internal saldırı türüne örnektir. Blackhole saldırısı DoS saldırısıdır. Saldırı bilgi güvenliğinin temel başlıklarından erişilebilirlik ve bütünlük başlıklarını ihlal etmeye yönelik yapılmaktadır. 79 no lu düğüm ile 100 no lu düğüm arasında trafik 60, 61, 62, 63, 33, 74 no lu saldırgan düğümler tarafından engellenmiştir. Saldırı 15,1 dakikada başlatılması tasarlanmıştır. Bu saldırıda zararlı düğümler iletilmek üzere kendine gelen trafiğin hepsini düşürerek iletişimi ilgili düğümler arasında engellemektedir. Selective Forwarding saldırısının aksine kendisine gelen tüm trafik paketlerini düşürmektedir. Çalışmadaki son saldırı olan Flooding saldırısı, DDoS saldırısı olup bu saldırı bilgi güvenliğinin temel öğelerinden olan ulaşılabilirliği ve kullanılabilirliği ihlal etmek için gerçekleştirilmektedir. Çalışmada Flooding saldırısı 72,1 dakikada başlayarak 18 no lu düğüm ile 36, 37, 38, 39, 40 no lu düğümler arasında gerçekleştirmesi kodlanmıştır. Saldırı

gerçekleştirirken RREQ paketleri kullanılmasına rağmen herhangi bir iletişim amacı bulunmamaktadır. Trafik yoğunluğu bakımından ağı işgal ederek, normal iletişimi geciktirmeye ya da düşürmeye diğer taraftan da düğümlerin enerjisini bitirmeyi hedeflenmektedir. Bu saldırı gerçekleştirilirken ilgili düğümler arasında normal iletişim kurmaksızın sadece istek paketleri gönderilmiştir. Bu yüzden yukarıda belirtilen düğümler arasında sadece istek paketlerinin oluşturduğu trafik ile saldırı gerçekleştirilmiştir. Saldırıların yanı sıra normal trafiğe de ihtiyaç duyulmaktadır. KSA'nın günlük uygulamalarında da saldırı gerçekleştirilirken aynı zamanda normal ağ trafiği de devam etmektedir. Ayrıca öğrenme modellerinin veriyi kategorize ederken normal ağ trafik verisine de ihtiyaç duymaktadır. Bu yüzden çalışmada normal trafiğe de yer verilmiştir. Normal ağ trafiği 160 no lu düğüm ile 137 no lu düğümden 199 no lu düğüm arasındaki tüm düğümler ile gerçekleştirilmiştir. Veri seti Blackhole, Flooding, Selective Forwarding saldırı trafiklerini içerdiğinden dolayı veri setinin adı saldırıların baş harflerinin alfabetik olarak sıralanması ile oluşturulmuştur. Oluşturulan veri seti Wireless Sensor Network Blackhole, Flooding, Selective Forwading (WSN-BFSF) şeklinde adlandırılmıştır. Senaryonun zaman akış şeması Şekil 7.5'de gösterilmiştir.



Şekil 7.5. Senaryo akış şeması

WSN-BFSF veri setinde dengeli dağılım olması ve belirlenen senaryonun benzetim ortamında çalışmaya devam etmesi için belirlenen süreler ve düğümler ve özellikle seçilerek senaryo gerçekleştirilmiştir. Kısacası saldırı senaryo trafiği; gönderilen paket boyutu, iletişim zaman aralıkları, saldırının başlayacağı süreler, iletilen düğümde bekletilecek olan paket sayısı, düğüm enerji değeri, düğümler arasındaki uzaklık gibi değerler dikkate alınarak

oluşturulmuştur. Senaryonun tasarlama aşaması tamamlanarak sonraki aşama olan NS-2 platformunda çalıştırma aşamasına geçilmiştir. Tcl script dosyası Ubuntu terminalinde ns komutu ile çalıştırılmasının ardından .nam ve .tr (trace) uzantılı dosyalar otomatik olarak üretilmiştir. Bu dosyalardan .nam dosyası, düğümlerin simülasyon ara yüzünü çalıştırmak ve görüntülemek için üretilmektedir. Nam dosyası açılmadan simülasyon arayüzü görüntülenememektedir. Simülasyon ortamının çalıştırıldığında nam dosyası ile paralel trace (.tr) dosyası üretilmektedir. Ancak trace dosyasının içeriği uygulamanın tamamlanmasının ardından tamamlanmaktadır. Ağ çalışması sırasında kaynaklanacak herhangi bir hata durumunda ya da .nam ara yüzünün kapatılması durumunda trace dosyası o ana kadar gerçekleştirilen trafik bilgilerini kaydetmektedir. Trace dosyası, ağ trafiği, ağ paketleri, düğümler hakkında detaylı bilgiler içermektedir. NS-2 platformunun tüm versiyonlarında ağ trafiğini incelemek isteyen geliştirici bu çalışmada da olduğu gibi trace dosyası ile işlem yapmaktadır. Dosya boyutu, kullanılan düğüm sayısı ağ trafik yoğunluğu, ağın çalışma süresi, trafik türü gibi değişkenlere bağlı olarak değişiklik göstermektedir. Ağ trafiği çalışmasını hatasız bir şekilde tamamladıktan trace dosyasını üretilmektedir. Senaryoda üretilen trace (.tr) dosyası uygulamanın ham veri setidir. Dosya satırlar ve özelliklerden(sütunlardan) oluşmaktadır. Satırların içeriğine göre özellik(sütun) sayısı farklılık göstermektedir. Dosyada sözel kısaltmalar ve kısaltmaların aldıkları değerler bulunmaktadır. Bu dosya uygulamanın ham verisini oluşturmaktadır. Genel olarak trace dosyasının içeriği üç çeşit satırdan oluşmaktadır. Aşağıda çalışmadan alınmış satır örneklerine yer verilmiştir.

Düğüm enerji satırı:

N -t 0.100115 -n 78 -e 299.979723

Yukarıda belirtildiği gibi satırlar, değerlerin açıklamasını içeren sözel kısaltmalar ve devamında ise ilgili değeri içermektedir. Yukarıdaki satır toplam dört değişken içermektedir.

İstek trafik satırı:

s -t 0.100115000 -Hs 79 -Hd -2 -Ni 79 -Nx 1950.00 -Ny 720.00 -Nz 0.00 -Ne 300.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 4f -Mt 800 -Is 79.255 -Id -1.255 -It AODV -Il 106 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 100 -Pds 0 -Ps 79 -Pss 4 -Pc REQUEST

Yukarıdaki satırda ise 31 değişken ve bu değişkenlerin aldığı değerler bulunmaktadır.

Normal trafik satırı:

r -t 0.100000000 -Hs 79 -Hd -2 -Ni 79 -Nx 1950.00 -Ny 720.00 -Nz 0.00 -Ne 300.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 79.0 -Id 100.0 -It cbr -Il 512 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0

Yukarıdaki satırda ise 26 değişken ve bu değişkenlerin aldığı değerler bulunmaktadır. WSN-BFSF veri setinin işlenmemiş halindeki satırlar incelendiğinde satırların büyük bölümünü düğüm enerji satırından oluştuğu görülmektedir. Bu satırlar sadece durum, zaman, düğüm numarası ve düğüm enerji değeri bilgilerini içerdiği görülmektedir. Bu bilgiler trafiğin incelenmesi ve sınıflandırılması için yeterli değildir. Ayrıca yukarıda belirtilen üç tür satırın özellik sayıları farklıdır. Bu yüzden sütun eşitleme işlemi uygulanmıştır. Sütun eşitleme işlemi yapılırken sütun sayısı eşitlenecek değerden az olan satırların eksik sütunları sıfır değeri verilerek eşitlenmektedir [33]. Düğüm enerji satırları daha az sütuna sahip olduğu için durum, zaman, düğüm numarası ve enerji değeri sütunlarının dışında eksik olan sütunlara 0 değeri verilerek eşitleme sağlanmıştır [33]. Ancak bu satırlar 4 sütundan oluştuğundan dolayı kalan sütun değeri 0 verildiğinde veri setinin büyük bölümü 0 sayısından oluştuğu görülmektedir. Bu halde veri seti çeşitli öğrenme modelleri ile denemiş ancak objektif ve kesin sonuca ulaşılamayıp modelin hata verdiği görülmüştür. Ayrıca ilgili satırlarda 0 değerine sahip sütunların veri setinin genel değer ortalamasını büyük oranda etkileyerek yanılttığı görülmüştür. Düğüm enerji satırları çıkarıldığında bu satırlardaki bilgiler (durum bilgisi, düğüm numarası, enerji miktarı) diğer satırlarda da bulunduğu için öğrenme modelleri ile incelenirken herhangi bir bilgi eksikliğine sebep olmayacağı görülmüştür. Kalan diğer satırlarda (Normal trafik satırı, istek trafik satırı) sütun eşitleme işlemi yapılırken iki satır türünde bulunan tüm sütunlar alınıp incelenmiştir. İki satırda da ortak olan sütunlar incelenirken değeri değişmeyen ve sonuca etki etmeyen sütunlar çıkarılmıştır. Tüm satırlarda sabit aynı değeri alan ya da büyük oranda sabit değer alıp ortalamayı değiştirmeyecek sayıda farklı değer alan sütunlar çıkarılmıştır. Ortak sütunlarda bu şekilde işlemler ile sadeleştirme yapılmıştır. Diğer taraftan istek satırında olup da normal trafik satırında olmayan sütunlarda literatürdeki diğer çalışmalarda yapılan eksik sütunlara 0 değeri ile eşitleme işlemi uygulanmıştır. Bunlara ek olarak farklı boyutlardaki satırlar bir araya getirilirken benzer sütunlar farklı sıralarda yer aldığı için ilgili özelliklerin yerlerinde değişiklik yapılarak aynı sütunda olmaları sağlanmıştır. Ayrıca farklı satırların sütun sayısı farklı olmasından dolayı ardışık sütunların yerleri farklı sıradan başlamaktadır. Bunun

çözümü olarak farklı satırlardaki aynı sütunların alt alta gelmesi için ilgili boş sütunlara 0 değeri verilmiştir. Veri ön işleme aşamaları EK-1'de gösterilmiştir.

WSN- BFSF veri seti şekil EK-1'deki ön işlemlerin ardından öğrenme modeller ile analiz edilmeye hazır hale getirilmiştir. Kullanıma hazır hale gelen veri seti 16 özellik boyutunda ve 312 106 satırdan oluşmaktadır. Veri setini hazır hale getirilme aşamaları EK-1'de veri seti elde etme aşamaları şemasında gösterilmiştir.

8. WSN-BFSF VERİ SETİ ÖZELLİKLERİ

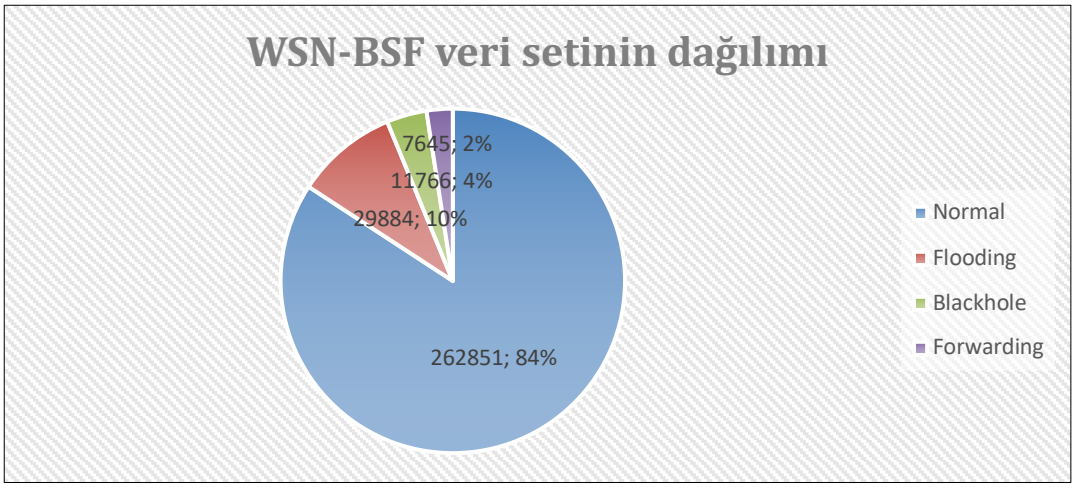
Ham veri setinde gerekli ön işlemler gerçekleştirildikten sonra hazır hale getirildiğinde 16 özellikli ve 312 106 satırlı bir veri seti elde edilmiştir. Veri seti Blackhole, Flooding, Selective Forwarding saldırı trafikleri ve Normal trafik olmak üzere 4 farklı trafikten oluşmaktadır. Elde edilen veri setinin özellikleri ve açıklaması Çizelge 8.1'de yer verilmiştir.

Çizelge 8.1. WSN-BFSF veri setinin özellik açıklaması

No	Uygulamada kullanılan 16 özellik listesi	Kısaltmaların açıklaması
1	Event	Trafik üzerinde gerçekleştirilen işlem hakkında bilgi içerir. s için 1, r için 2, f için 3, d için 4, N için ise 5 değeri ile temsil edilmiştir.
2	Time	Satırda gerçekleştirilen olayın zamanı
3	S_Node	Kaynak düğüm numarası
4	Node_id	İlgili düğümün düğüm numarası
5	Rest_Energy	İlgili düğümün kalan enerji miktarı
6	Mac_Type_Pck	Paketin MAC çeşidi
7	Source_IP_Port	Kaynak düğümün port numarası
8	Packet_Size	İletilen paket boyutu
9	TTL	İletilen trafiğin ağdaki ömrü
10	Hop_Count	Geçilen düğüm sayısı
11	Broadcast_ID	Yayın paketlerinin id numarası
12	Dest_Node_Num	Hedef düğümün id numarası
13	Dest_Seq_Num	Hedefe iletilen trafiğin sıra numarası
14	Src_Node_ID	Kaynak düğümün id Numarası
15	Src_Seq_Num	Hedefe iletilen trafiğin kaynak sıra numarası
16	Sınıf	Sınıflandırılan ağ trafiğin türü

Özelliklerin ilki olan Event özelliği o satırda hangi işlemin (alma, gönderme, düşürme, iletme) gerçekleştiğini sözel olarak ifade etmektedir. Time özelliği satırda belirtilen olayın gerçekleşme zamanını belirtmektedir. S_Node özelliği kaynak düğümün düğüm numarasını ifade etmektedir. Bu özellik trafiği oluşturan düğüm hakkında bilgi vermektedir. Node_id özelliği satırdaki olayın gerçekleştiği düğüm hakkında bilgi vermektedir. Rest_Energy özelliği ise satır hangi düğümden bahsediyor ise o düğümün kalan enerji miktarını göstermektedir. Mac_Type_Pck özelliği paketin mac çeşidi hakkında bilgi vermektedir. 0-800 değerlerini almaktadır. Source_IP_Port özelliği kaynak düğümün port numarası

hakkında bilgi vermektedir. Packet_Size özelliği düğümler arasında iletilen paketlerin boyutları hakkında bilgi vermektedir. TTL özelliği düğümler arası paketler iletilirken loopa girmesi durumunda ağı işgal etmektedir. Böyle durumları engellemek için paketin iletilebileceği düğüm sayısı belirlenerek gereksiz trafiklerin önüne geçilmektedir. Böylelikle paketler sonsuza kadar iletilmemekte TTL değerine ulaştığı an paket düşürülmektedir. Hop_Count özelliği ise belirtilen satıra kadar paketin kaç düğüme ulaşıp, iletildiğini ifade etmektedir. Broadcast_ID özelliği genellikle istek paketleri gibi tüm ağa iletilen paket trafiğini ifade eden satırlarda bulunan bir özelliktir. Yayın paketinin numarasını ifade etmektedir. Dest_Node_Num özelliği paketin (trafiğin) iletileceği hedef düğümün numarasını belirtmektedir. Dest_Seq_Num özelliği KSA'da verinin tazeliğini belirten paket sıra numarasını ifade etmektedir. İletilen paketlerin sıra numarası verinin güncelliği hakkında bilgi vermektedir. Src_Node_ID özelliği trafiği üreten kaynak düğüm hakkında bilgi vermektedir. Src_Seq_Num özelliği hedefe gönderilen veri bir sıra halinde iletilmektedir. Bu bakımdan verinin iletilme sırası önem arz etmektedir. Veri setindeki 16. özellik (son) trafiğin sınıfını belirtmektedir. Veri seti literatürde ve bu çalışmada gözetimli öğrenme ile incelenirken trafiğe ait sınıf etiketi öğrenmenin doğru olabilmesi için önem arz etmektedir. WSN-BFSF veri setinin kullanıma hazır hale getirildiğinde her trafik türünden farklı sayılarda veri elde edilmiştir. WSN-BFSF veri setinde normal trafiğin saldırı trafiklerine göre sayısının fazla olduğu görülmektedir. Bu kullanılacak modellerde normal trafiğin öğrenilmesi açısından ve yanlış sınıflandırmanın önüne geçilmesi açısından önem arz etmektedir. WSN-BFSF veri setinin trafik çeşidine göre dağılımı Şekil 8.1'de yer almaktadır.



Şekil 8.1. WSN-BFSF veri seti dağılımı

9. MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ANALİZLERİ

Tez çalışmada KSA'da DoS saldırılarının tespiti için yeni bir veri kümesi oluşturulmuştur. Oluşturulan veri kümesi WSN-BFSF olarak adlandırılmıştır. WSN-BFSF veri kümesi Blackhole, Flooding, Selective Forwarding olmak üzere üç farklı DoS atağı ve normal trafik örneklerinden oluşmaktadır. Çizelge 9.1'de oluşturulan veri kümesinden örnek kayıtlar gösterilmektedir.

Çizelge 9.1. WSN-BFSF veri dağılımı

Sınıf	Satır sayısı	Yüzdelik dağılım
Normal	262 851	84,22
Flooding	29 844	9,56
Blackhole	11 766	3,77
Forwarding	7 645	2,45
Total	312 106	100

Veri kümesi Çizelge 9.1'de gösterildiği gibi bu dört farklı sınıf etiketine sahip toplamda 312 106 örnekten oluşmaktadır.

9.1. Veri Ön İşleme

Çalışmanın 7. ve 8. bölümlerinde veri setinin elde edilme ve verilerin modellerin anlayacağı değerlere dönüştürülme aşamaları anlatılmıştır. Ham haldeki veri seti ayıklama ve dönüştürme işlemlerinden sonra WSN-BFSF veri seti modeller ile incelenmeye hazır hale getirilmiştir. Veri seti modeller ile incelenip sonuçlar üretilmeden önce üç aşamalı ön işlem gerçekleştirilmiştir. İlk aşamada veri setinin son sütununda ilgili satırın hangi trafik türüne ait olduğunu belirten sözel ifadeler (Blackhole, Flooding, Selective Forwarding ve Normal) sayısal değerlere çevrilmiştir. Sözel ifadelerin bulunduğu bu sütuna trafik sınıf (class) sütunu olarak adlandırılmaktadır. Sınıf sütunundaki bu sözel ifadeler ilgili satırdaki trafiklerin etiketleri (label) olarak adlandırılmaktadır. Modeller ile sınıflandırma yapılırken bu etiketler önem arz etmektedir. Özellikle gözetimli öğrenmede etikete (label) göre sınıflandırma işlemi gerçekleştirilmektedir. Ancak trafik sınıf sütununda Blackhole, Flooding, Selective Forwarding ve Normal gibi sözel etiketler yer almaktadır. Bu ifadeler modeller için bir anlam ifade etmemektedir. Bu yüzden bu sözel etiketlere sayısal dönüşümler uygulanmıştır. Çizelge 9.2'de sayısal dönüşüm (Label Encoding) sonuçlarına yer verilmiştir.

Çizelge 9.2. Sayısal dönüşüm işlemi

Aldığı Değer	Trafik Türü
0	Blackhole
1	Flooding
2	Selective Forwarding
3	Normal

One-Hot Encoding işlemi ile veri kümesindeki kategorik değerler sayısal değerlere atanmaktadır. Bu işleme Label Encoding denilmektedir. Etiketin kodlanması (Label Encoding) aşaması Google Collab. ortamında Scikit-Learn kütüphanesinden Label Encoding metodu ile gerçekleştirilmiştir. Class sütunundaki sözel (string) değerler sayısal (integer) değerlere dönüştürülmüştür. Blackhole saldırısının sınıf etiketine 0 değeri, Flooding saldırısının sınıf etiketine 1 değeri, Selective Forwarding saldırısının sınıf etiketine 2 değeri ve Normal trafiğin sınıf etiketine 3 değeri verilmiştir.

İkinci aşamada özellik azaltma işlemi gerçekleştirilmektedir. Modellerin kullanımına hazır hale getirilen veri seti ile denemeler yapıldığında özelliklerin bazılarının tüm satırlarda aynı değeri aldığı bundan dolayı sonuca bir katkı sağlamadığı görülmüştür. Ayrıca bazı sütunlarda her ne kadar farklı değerler görülse de çoğunluğu 0 değerinden oluştuğundan modeller ile incelendiğinde hata mesajlarının alındığı görülmüştür. Bu yüzden veri setinin bazı sütunlarına filtre işlemi uygulanarak veri seti 16 sütuna (özelliğe) indirgenmiştir. İndirgeme işlemlerine ve özelliklerin açıklamalarına 7. bölümde yer verilmiştir. İlgili özelliklerin seçimi yapılırken öncelikle Google Collab. ortamında veri seti bir değişkene tanımlanmıştır. Tanımlamanın ardından değişken üzerinden select metodu ile özellik seçim işlemi gerçekleştirilmiştir.

Özellik seçiminin ardından üçüncü aşama olan veri seti üzerinde normalizasyon işlemi gerçekleştirilmektedir. Normalizasyon işleminin amacı aynı özelliğe sahip farklı satırların ilgili özelliklerinin aldıkları değer aralığının büyük olması durumunda bu değerleri 0 ile 1 arasına, sahip oldukları değerlerine göre yerleştirmektir. Değer aralığının büyük olması sonucun büyük değerler tarafından belirlenmesine neden olmaktadır. Oysaki her satırın aynı özelliği farklı oranlarda sonuca katkı sağlamaktadır. Ancak büyük değerlere sahip satırların sonucu dengesiz oranda değiştirmesi doğru tespit oranını etkilemektedir. Bu yüzden normalizasyon işlemi her satırdaki aynı özelliğin değerleri arasında büyük farklılıkları belirli

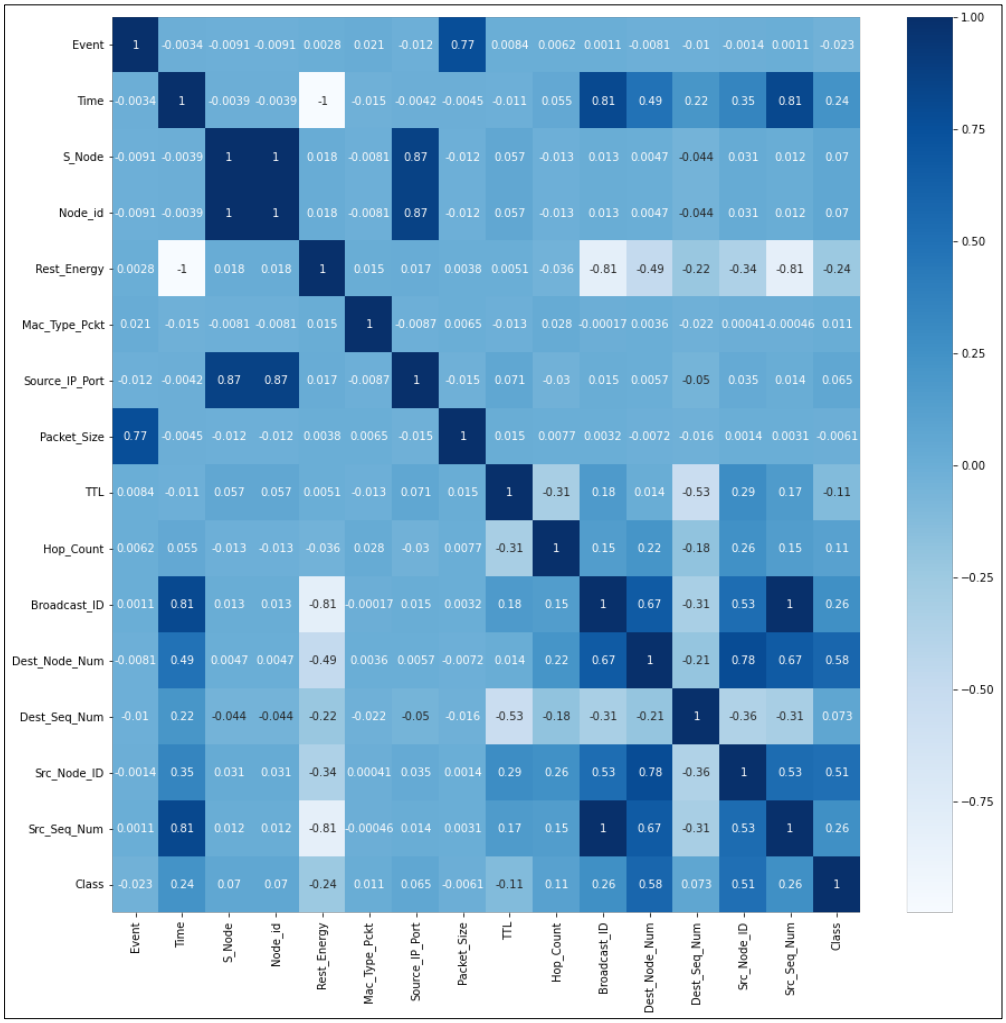
bir aralığa sığdırarak sonuca etkisini sınırlamaktadır. Eş. 9.1'de gösterilen normalizasyon işlemi gerçekleştirilmektedir.

$$x' = \frac{x-\mu}{\sigma} \tag{9.1}$$

Eş. 9.1'de x orijinal değeri x′ normalize edilmiş değeri μ ve σ ise sırasıyla ortalama ve standart sapma değerlerini ifade etmektedir. Bu çalışmada özelliklerin normalizasyon işlemi MLib kütüphanesin de normalizer aracı import edilerek gerçekleştirilmiştir. Komutun kullanımda normalizer metodu ve veri setinin atandığı değişken kullanarak normalizasyon işlemi gerçekleştirilmiştir.

9.2. Analiz İşlemleri

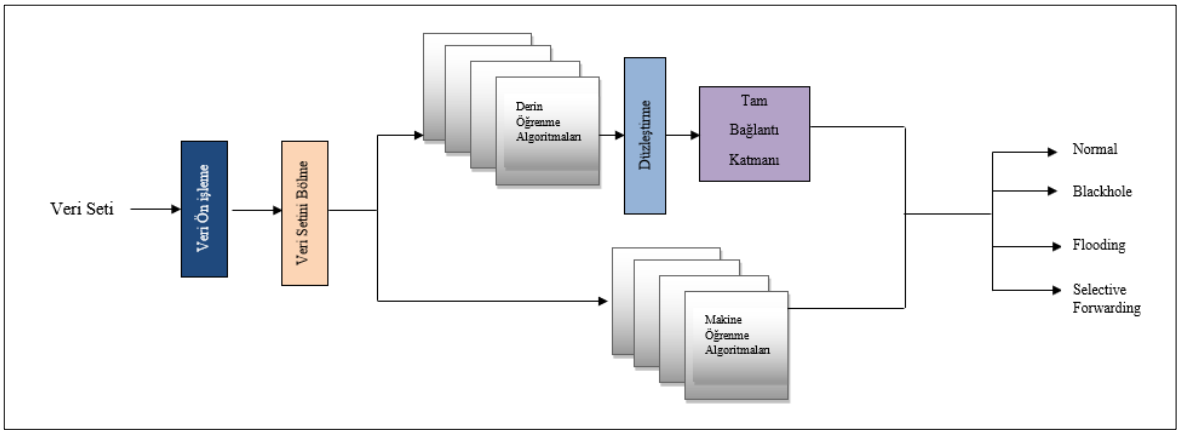
Çalışmada oluşturulan veri kümesinde Bölüm 8'de detaylı olarak anlatıldığı gibi 16 adet özellik bulunmaktadır. Veri kümesindeki özellikler arasındaki korelasyon Pearson Correlation Coefficient kullanılarak Şekil 9.1'de gösterilmiştir.



Şekil 9.1 WSN-BFSF veri seti için Pearson Correlation Matrix sonuçları

Şekil 9.1'de WSN-BFSF veri kümesindeki her bir özelliğin veri seti içindeki diğer özelliklerle ilişkilerini gösteren özellik analizi sunulmuştur. Pearson Correlation Matrix özellikler arasında ilişkilerin derecesini vermektedir. Bu ilişkilerde 1 en yüksek ilişkiyi -1 ise en düşük ilişkiyi temsil etmektedir. Şekil 9.1'deki diagonal eksene bakıldığında aynı özellikler bu eksende kesiştiği için ilişki değerleri en yüksek değer olan 1 değerini almıştır. İlişki değerleri +1 ile -1 değer aralığına dağıtılmıştır. Aldıkları ilişki değerlerine göre koyu maviden beyaz renge kadar ilişki değerleri renkler ile temsil ettirilmiştir. İlişki değeri 1 ise koyu mavi renk ile gösterilmiş, ilişki değeri -1 ise beyaz renk ile gösterilmiştir.

Bu bölümde, DoS tespiti için oluşturulan veri kümesinin sınıflandırma performansları çeşitli makine öğrenmesi ve derin öğrenme algoritmaları kullanılarak karşılaştırılmıştır. Sınıflandırma süreci özet olarak Şekil 9.2'de gösterilmektedir.



Şekil 9.2. Makine/derin öğrenme akış şeması

Şekil 9.2'de görüldüğü gibi kullanıma hazır hale getirilen veri seti ön işlemden (etiketleme, özellik seçimi ve normalizasyon) geçirilmiştir. Sonraki aşamada veri seti test ve eğitim seti olmak üzere ikiye ayrılmıştır. Ayrılan veri setinin eğitim bölümü 4 farklı derin öğrenme ve 4 farklı makine öğrenme modelinin eğitim aşamasında kullanılmıştır. Test seti ise modellerin öğrenmesinin başarılı olup olmadığını tespit etmek için kullanılmıştır. Çalışmalar, Google Colab ortamında Apache Spark büyük veri platformu üzerinde Python programlama dili desteği sağlayan Pyspark aracı kullanılarak yapılmıştır. Makine öğrenmesi ve derin öğrenme algoritmaları için PySpark MLib kütüphanesi içinde yer alan sırasıyla Scikit-Learn ve Keras kütüphaneleri kullanılmıştır. Önerilen yöntem, 8 farklı makine öğrenmesi ve derin öğrenme algoritması ile karşılaştırılarak değerlendirmeler yapılmış ve sonuçlar yorumlanmıştır.

Sonuçların değerlendirilmesinde doğruluk, hassasiyet, F-skoru, Recall, ROC ve Precision-Recall eğrileri gibi literatürde en yaygın kullanılan parametreler kullanılmıştır. Bu parametreler hata matrisi verilerinden elde edilmektedir. Hata matrisinin temel elemanları doğru-pozitif (TP), doğru-negatif (TN), yanlış-pozitif (FP) ve yanlış-negatif (FN)'tir. TP, doğru bir şekilde saldırı olarak sınıflandırılan örnek sayısını temsil etmektedir. TN, doğru bir şekilde normal olarak sınıflandırılan örnek sayısını temsil etmektedir. FP, normal örneklerin saldırı örnekleri olarak yanlış sınıflandırılmasını ifade etmektedir. Benzer şekilde, FN, saldırı örneklerinin yanlış sınıflandırılarak normal örnekler olarak kabul edilmesini ifade etmektedir. Doğruluk parametresi, doğru sınıflandırılmış tüm örneklerin (TP, TN) tüm örneklere (TP, TN, FP ve FN) oranı olarak tanımlanmaktadır. Eş. 9.2'de gösterilmektedir. Hassasiyet, doğru sınıflandırılmış tüm saldırıların (TP) doğru sınıflandırılmış tüm saldırı (TP) ve yanlış sınıflandırılmış normal örneklerin (FP) sayısına oranıdır. Hassasiyet, Eş.

9.4'te gösterilmektedir. Recall doğru sınıflandırılan pozitif örneklerin sayısının doğru olarak sınıflandırılmış tüm örneklerin sayısına oranını ifade etmektedir ve Eş. 9.5'te gösterilmektedir. Hassasiyet ve Recall parametrelerinin harmonik ortalaması F-skoru olarak bilinir ve Eş. 9.3'de gösterilmektedir.

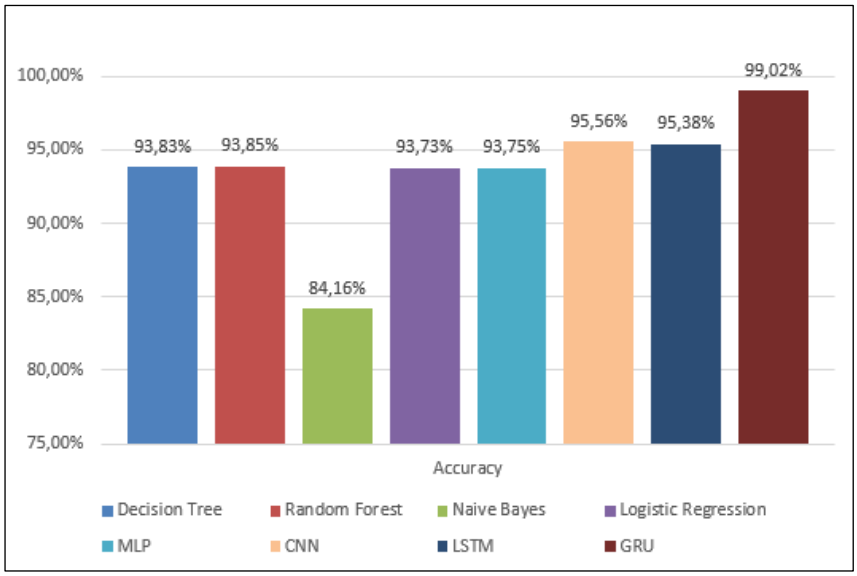
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9.2}$$

$$F - Score = \frac{2TP}{2TP+FP+FN} \tag{9.3}$$

$$Precision = \frac{TP}{TP+FP} \tag{9.4}$$

$$Recall = \frac{TP}{FN+TP} \tag{9.5}$$

Şekil 9.3'te WSN-BFSF veri kümesi üzerinde literatürde sıklıkla kullanılan çeşitli makine öğrenmesi ve derin öğrenme algoritmalarının doğruluk parametresine göre sınıflandırma sonuçlarının karşılaştırmalarını sunmaktadır.



Şekil 9.3. Modellerin doğruluk yüzde sonuçları

Şekil 9.3'de görüldüğü üzere atakların sınıflandırılmasında GRU derin öğrenme algoritması %99,02 doğruluk değeri ile diğer yöntemlere göre en iyi doğruluk sonucunu vermektedir. Bu çalışmada, veri kümesi eğitim için %70 ve test için %30 olmak üzere ikiye bölünmüştür. Kullanılan derin öğrenme algoritmalarındaki hiperparametreler için öncelikle genel geçer değerler verilmiş ve en iyi sonuçlar için tuning işlemi yapılmıştır. Veri setinin eğitim ve test

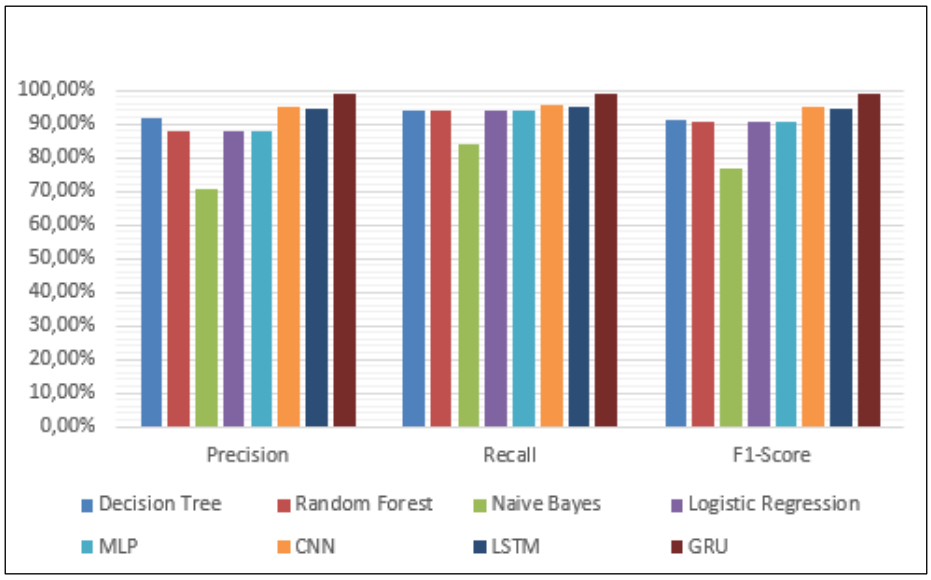
seti şeklinde bölünmesi ve eğitimi için ayrı, test için ayrı setlerin kullanılması veri setinin geçerliliği açısından önem arz etmektedir [33]. Çalışmada veri setinin %70'i modellerin eğitimi için kullanılmıştır. Veri setinin kalan %30'u test aşamasına kadar hiçbir şekilde kullanılmamıştır. Yani model test aşamasında hiç bilmediği bir veriyi (veri setinin %30'luk bölümünü) yüksek doğruluk yüzdesi ile sınıflandırmayı başarmıştır. Şekil 9.3'te ve Şekil 9.4'te yer verilen sonuçlar WSN-BFSF veri setinin geçerliliğini göstermektedir. Literatüre bakıldığında veri setlerinin üretildiği [33] ve [42] çalışmalarda da aynı tekniklerle veri setlerinin geçerlilikleri göstermişlerdir. Derin öğrenme algoritmaları olarak birçok sınıflandırma probleminde yaygın olarak tercih edilen MLP, CNN, LSTM ve GRU kullanılmıştır. Makine öğrenmesi algoritmaları olarak ise literatürde oldukça yaygın kullanılan temel makine öğrenmesi algoritmaları kullanılmıştır. Bu algoritmalar, Random Forest, Decision Tree, Naive Bayes ve Logistic Regression'dur.

GRU algoritması için elde edilen sonuç tuning işlemi sonucunda Çizelge 9.3'te gösterilen hiperparametreler kullanılarak elde edilmiştir.

Çizelge 9.3. GRU algoritması ile kullanılan hiperparametreler

Hyperparameters	Values
Activation function	Relu, Softmax
Number of Epoch	30
Learning rate	0,01
Output size	150
Drop rate	0,01
Units	128
Optimizer	Adam
Loss	Categorical entropy
Hidden layer	1

Şekil 9.4'te ve Çizelge 9.4'te accuracy, precision, recall ve F1-Score parametrelerine göre sınıflandırma algoritmalarının performanslarının karşılaştırması sunulmuştur.



Şekil 9.4. Precision, Recall and F1-Score parametrelerine göre karşılaştırma sonuçları

Bu parametrelerin sonuçları incelendiğinde önerilen yöntem her bir parametre için de en iyi sonuçları sergilemektedir.

Çizelge 9.4. Accuracy, Precision, Recall ve F-Score sonuçları

Algoritmalar	Precision	Recall	F-Score	Accuracy
Decision Tree	0,920382	0,937973	0,910834	0,937973
Random Forest	0,878815	0,938451	0,907187	0,938451
Naive Bayes	0,708250	0,841576	0,769178	0,841576
Logistic Regression	0,878815	0,937292	0,907112	0,937292
MLP	0,878815	0,937451	0,907187	0,937451
CNN	0,951428	0,955641	0,948956	0,955641
LSTM	0,946327	0,953759	0,946692	0,953759
GRU	0,990003	0,990182	0,989900	0,990182

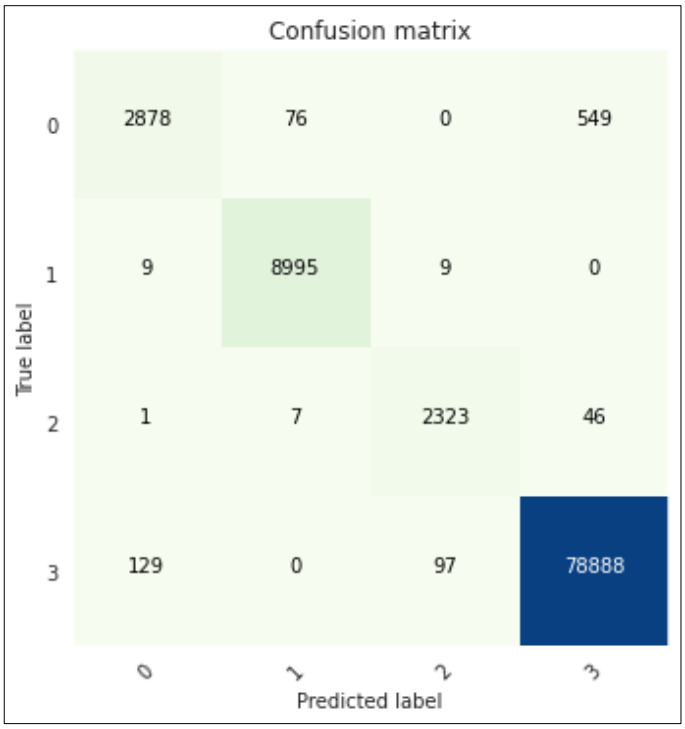
Bunun yanı sıra Çizelge 9.5'te GRU derin öğrenme algoritmasının değerlendirme parametrelerine göre detaylı sınıflandırma sonuçları sunulmuştur.

Çizelge 9.5. GRU algoritmasının sınıflandırma performansı

Parametreler	Blackhole	Flooding	Selective Forwarding	Normal
Accuracy	0,82158150	0,99800288	0,97728229	0,99714336
Precision	0,95392774	0,99085702	0,95636064	0,99251412
Recall	0,82158150	0,99800288	0,97728229	0,99714336
F1-Score	0,88282209	0,99441711	0,96670828	0,99482336

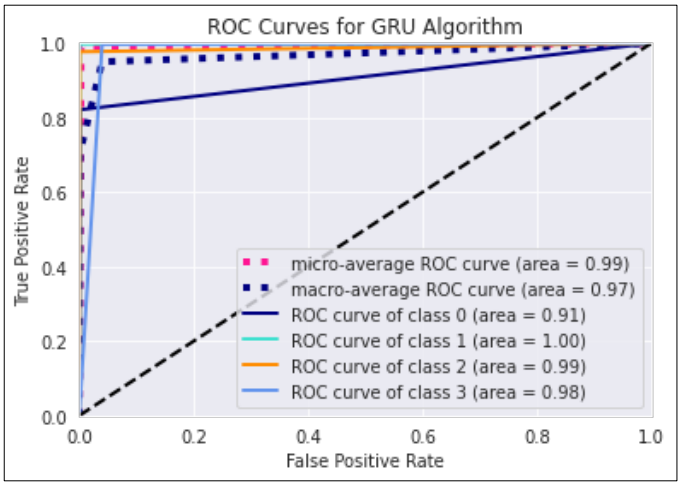
Elde edilen sonuçlardan GRU derin öğrenme algoritması Blackhole, Flooding, Selective Forwarding saldırı trafikleri ve Normal trafik sınıflarına göre sırasıyla %82,16, %99,80, %97,73 ve %99,71 sınıflandırma doğruluğu elde etmiştir.

Önerilen algoritmanın her bir sınıfa ait sınıflandırma sonuçları Confusion Matrix üzerinde Şekil 9.5'te gösterilmektedir.



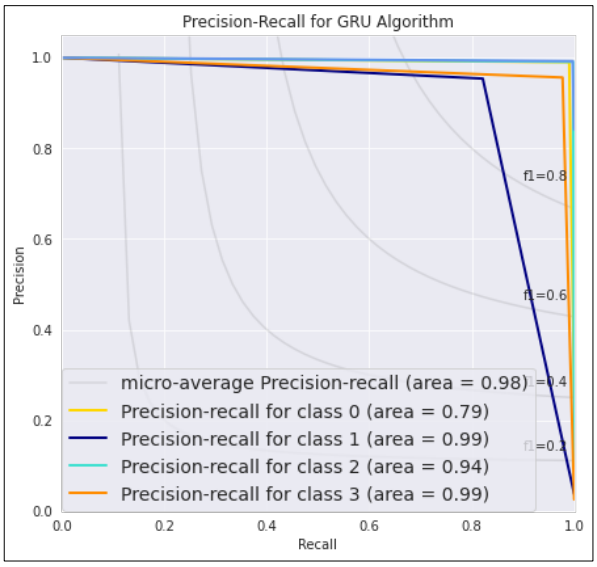
Şekil 9.5. Confusion Matrix

Confusion Matrix'ten önerilen algoritmanın tüm sınıflar için başarılı olduğu görülmektedir. Şekillerde sayısal olarak gösterilen sınıf numaralarının isim eşleşmeleri Çizelge 9.2'de gösterilmiştir. Şekil 9.5'te veri setinin trafik sınıflarına göre gerçek dağılımı ile modellerin sınıflandırdığı dağılım görülmektedir. Düşey eksende veri setinin gerçekte hangi trafikten kaç adet örnek içerdiğinden gösterilmektedir. Yatay eksende ise modellerin tahminleri yer almaktadır. Bunun yanı sıra, imbalanced veri kümelerindeki sınıflandırma performanslarının doğru karşılaştırması için kullanılan ROC ve Precision-Recall eğrileri Şekil 9.6'da ve Şekil 9.7'de sırasıyla gösterilmektedir.



Şekil 9.6. ROC eğrisi

Çalışmada sınıflandırma başarısının kalitesinin ölçülmesinde kullanışlı olan bir diğer parametre ROC eğrileri de Şekil 9.6'da elde edilmiştir. ROC eğrilerinde X ekseni False Positive oranını gösterirken Y ekseni True Positive oranını göstermektedir. Bundan dolayı, sol üst köşeye yakın sonuçlar ideal sonuçlar olarak değerlendirilmektedir. Çalışmada elde edilen sonuçlar değerlendirildiğinde GRU algoritması için eğrilerin ideal noktaya yakın olarak tüm sınıflar için oldukça başarılı sonuçlar elde edildiği görülmektedir.



Şekil 9.7. Precision-Recall eğrisi

Precision-Recall eğrileri, sınıflar çok dengesiz olduğunda tahmin başarısının ölçülmesinde kullanılan oldukça kullanışlı bir parametredir. Bu eğri precision ve recall arasındaki dengeyi göstermektedir. Eğrinin altındaki alanın yüksek olması hem yüksek recall hem de yüksek

precision'u temsil etmektedir. Şekil 9.7'de elde edilen Precision-Recall eğrileri incelendiğinde önerilen GRU algoritması için eğrilerin altında kalan alan maksimum değer olan 1'e yakın olarak tüm sınıflar için oldukça başarılı sonuçlar elde edildiğini göstermektedir. Çizelge 9.6'da sınıflandırma algoritmalarının her bir sınıf için doğruluk parametresine göre karşılaştırması sunulmuştur.

Çizelge 9.6. Algoritmaların trafik sınıflarına göre doğruluk değer performansı

Algoritmalar	Blackhole	Flooding	Selective Forwarding	Normal
Decision Tree	0,01	0,99	0,01	1,00
Random Forest	0,04	1,00	0,01	1,00
Naive Bayes	0,01	0,01	0,01	1,00
Logistic Regression	0,01	1,00	0,01	1,00
MLP	0,01	1,00	0,01	1,00
CNN	0,42	0,98	0,41	0,99
LSTM	0,35	0,98	0,65	0,99
GRU	0,82	0,99	0,97	0,99

Elde edilen sonuçlardan derin öğrenme algoritmalarının geleneksel makine öğrenme algoritmalarına göre daha iyi sonuçlar elde ettiği gözlenmiştir. En düşük performansı Naive Bayes algoritması göstermiştir. Bu çalışmada Multinominal Naive Bayes algoritması kullanılmıştır. Naive Bayes algoritması atak sınıflarına ait verileri yüksek oranda tespit edemediği için en düşük performansı göstermiştir. Çizelge 9.6 incelendiğinde makine öğrenmesi algoritmaları az sayıda veri içeren saldırı trafik sınıflarını doğru sınıflandıramadığı görülmektedir. MLP, CNN ve LSTM derin öğrenme algoritmaları ise Blackhole ve Selective Forwarding sınıflarına ait verileri başarılı bir şekilde sınıflandıramamaktadır. Bu durum veri kümesinin gerçeğe uygun olarak imblanced bir şekilde oluşturulmasından kaynaklanmaktadır. Gerçek WSN ortamlarında da ağdan geçen normal trafik miktarı saldırı trafik miktarına göre oldukça fazladır.

10. SONUÇ VE ÖNERİLER

Tez çalışmasında KSA'da sıklıkla görülen ağ katmanı saldırılarından Blackhole, Flooding, Selective Forwarding saldırıları NS-2 benzetim ortamında gerçekleştirilmiştir. Saldırıların çalıştırılabilmesi için protokol sisteminde bulunan aodv.h ve aodv.cc dosyalarında gerekli değişiklikler yapılmıştır. Senaryo oluşturulurken ilk aşamada NSG 2.1 aracı kullanılarak temel ağ mimari ve düğüm özellikler tanımlanmıştır. Sonraki aşamada ağ ve düğüm ile ilgili ihtiyaç duyulan değişiklikler tcl script dosyası üzerinde değişiklikler yapılarak gerçekleştirilmiştir. Aynı ağda üç farklı saldırı trafiği ve bir normal trafik olmak üzere toplam 4 farklı trafik türü oluşturulmuştur. WSN-BFSF veri seti kullanılırken normal trafik veri örneklerine de ihtiyaç duyulacağından saldırılar gerçekleştirilirken aynı zamanda normal ağ trafiği de çalışmaya devam etmiştir. NS-2 benzetim ortamında aynı ağda farklı trafikler çalıştırılırken trafik çakışması, aşırı trafik yoğunluğu, dengesiz miktardaki paket boyutu gibi etkenlerden dolayı sıklıkla hata mesajları üretilmekte ve platformun çalışmasını durdurmaktadır. İstenilen saldırı trafiğinin elde edilmesi için bu etkenlere dikkat edilmesi önem arz etmektedir. WSN-BFSF veri seti, 200 düğümün 1600 dakika boyunca çalışarak ürettiği ağ trafik verilerinden oluşmaktadır. İletişim protokolü olarak düz ağ mimarisine sahip uçtan uca haberleşme algoritması ile çalışan AODV protokolü kullanılmıştır. Ağ trafiği tamamlandıktan sonra WSN-BFSF veri setinin işlenmemiş hali benzetim platformundan alınarak aşamalı olarak ön işlemlerden geçirilmiştir. Ön işlem sonrasında kullanıma hazır hale gelen veri seti 16 özellik boyutunda ve 312 106 satırdan oluşmaktadır. WSN-BFSF veri setinin %84'ü normal trafik, %10 Flooding saldırı trafiği, %4 Blackhole saldırı trafiği ve %2 Selective Forwarding saldırı trafiğinden oluşmaktadır. WSN-BFSF veri seti; Random Forest, Decision Tree, Naive Bayes ve Logistic Regression makine öğrenme modelleri ve MLP, CNN, LSTM, GRU derin öğrenme modelleri olmak üzere toplamda 8 farklı model ile incelenmiştir. Saldırı ve normal trafiği sınıflandırmanın yanı sıra saldırı trafiğini de saldırı türüne göre ayrıca sınıflandırılmıştır. İncelenen veri seti sınıflandırılırken modeller; learning rate, epoch, drop rate optimizer ve loss fonksiyonu gibi hiperparametre ile kombinlenerek en yüksek doğruluk yüzdesi elde edilmiştir. %99,02 en yüksek doğruluk tespit oranına GRU derin öğrenme modeli ile ulaşılmıştır. Bu orana ulaşılırken GRU modeli ile ReLu ve Softmax aktivasyon fonksiyonları, 30 epoch değeri 0,01 learning rate, categorical entrophy loss fonksiyonu ve Adam optimizer algoritması kullanılmıştır.

KAYNAKLAR

1. Abazeed, M. et al. (2015). A review of secure routing approaches for current and next- generation wireless multimedia sensor networks. *International Journal of Distributed Sensor Networks*,11(10),1-22.

2. Ekong, V. and Ekong, U. (2016). A survey of security vulnerabilities in wireless sensor networks. *Nigerian Journal of Technology*, 35(2), 392.

3. Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, 1 (2-3), 293-315.

4. Deif, D. and Gadallah, Y. (2017). An ant colony optimization approach for the deployment of reliable wireless sensor networks, *IEEE Access*, 5 (1), 10744-10756.

5. Sheng, Z, Mahapatra C., Zhu, C. and Leung, V. (2015). Recent advances in ındustrial wireless sensor networks toward efficient management in IoT. *IEEE Access*, 3 (1), 622-637.

6. Selvi, M., Thangaramya K, Ganapathy, S., Kulothungan, K., Khannah Nehemiah H and Kannan A. (2019). An energy aware trust based secure routing algorithm for effective communication in wireless sensor networks. *Wireless Personal Communications*, 105 (4), 1475-1490.

7. Dener, M. (2014). Security analysis in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 10 (10),303-501.

8. Dener, M. (2018). A new energy efficient hierarchical routing protocol for wireless sensor networks. *Wireless Personal Communications*, 101(1), 269-286.

9. Meghdadi, M., Ozdemir, S., and Güler, I. (2008). Security in wireless sensor networks: problems and solutions. *International Journal of Information Technologies*, 1(1), 35–40.

10. Carman, D. W., Krus, P. S. and Matt, B. J. (2000). *Constraints and approaches for distributed sensor network security*, 3(1),38–45.

11. Meghdadi, M., Ozdemir, S., and Güler, I. (2008). Security in wireless sensor networks: problems and solutions. *International Journal of Information Technologies*, 1(1), 35–40.

12. Yong-Min, L., Shu-Ci, W., Xiao-Hong, N. (2009, 13-15 November). *The architecture and characteristics of wireless sensor network.* International Conference on Computer Technology and Development, Kota Kinabalu, 561-568.

13. Tseng, C., Wang, S., H. and Tsaur, W. (2015). Hierarchical and dynamic elliptic curve cryptosystem based self-certified public key scheme for medical data protection. *IEEE Transactions on Reliability*, 64 (3),1078-1085.

14. Singh, R. Singh, D., and Kumar, L. (2010). A review on security issues in wireless sensor network. *Journal of Information Systems Communications.* 1(1), 1-20.

15. Buratti, C., Conti, A., Dardari, D. and Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. *Sensors*, 9 (9),6869-6896.

16. Romer, K. and Mattern, F. (2004). The design space of wireless sensor networks. *IEEE Wireless Communucations*, 11(6), 54-61.

17. Tawalbeh, L. Hashish, S. and Tawalbeh, H. (2017). Quality of service requirements and challenges in generic WSN infrastructures. *Procedia Computer Science*, 109 (1), 1116-1121.

18. Yu, J., Lee, E., Oh, S., Seo Y. and Kim, Y. (2020). A survey on security requirements for WSNs: focusing on the characteristics related to security. *IEEE Access*, 8 (1), 45304-45324.

19. Vardhan, A., Hussain, M. and Garimella, R. M. (2016, December 23-25). *Simple and secure node authentication in wireless sensor networks.* International Conference and Workshops on Recent Advances and Innovations in Engineering, Surathkal,1-5.

20. Kim, J.M., Lee, H.S. J., Yi, J and Park, M. (2016). Power adaptive data encryption for energy efficient and secure communication in solar-powered wireless sensor networks, *Journal of Sensors*, 6(9),1-9.

21. Rajkumar, V. B. A., Rajaraman, G. and Chandrakanth, H. (2014). Security attacks and its countermeasures in wireless sensor networks. *International Journal of Engineering Research And Applications*, 4 (10). 4-15

22. Ferrari, P., Giorgi, G., Narduzzi, C., Rinaldi, S. and Rizzi, M. (2014). Timestamp validation strategy for wireless sensor networks based on IEEE 802.15.4 CSS. *IEEE Transactions on Instrumentation and Measurement*, 63(11). 2512-2521.

23. Yang, G., Dai, L., Si, G., Wang, S. and. Wang, S. (2019). Challenges and security issues in underwater wireless sensor networks. *Procedia Computer Science,* 147 (1), 210-216.

24. Ahmed, M., Huang, X., Sharma, D. and. Shutao, L. (2012). Wireless sensor network internal attacker identification with multiple evidence by dempstershafer theory. *Algorithms and Architectures for Parallel Processing*, 7440 (1),255-263.

25. Ahmed, M., Huang, X. and Cui, H. (2013). A novel two-stage algorithm protecting internal attack from WSNs. *International Journal of Comupuer Network Communications*, 5 (1). 97-116.

25. Sabeel, U. and Maqbool, S. (2013). Categorized security threats in the wireless sensor networks: Countermeasures and security management schemes. *International Journal of Computers and Applications*, 64(16), 19-28.

26. Tang J., Fan, P. and Tang X. (2007, October 31- November 3). A RSSI-based c ooperative anomaly detection scheme for wireless sensor networks. *International Conference of Wireless Communication Network Mobile Computers*, Kunming, 2783-2786.

27. Mandal, P. S. and. Ghosh, A. K. (2011, September 21-23). *Secure position verification for wireless sensor networks in noisy channels.* International Conference of Ad-Hoc Networks, Paris, 150-163.

28. Elleuchi, M., Cheikhrouhou, O., Obeid, A. M. and Abid, M. (2016). *An efficient secure scheme for wireless sensor networks.* International Conference of Security Information Network, 129-132.

29. Azeem, M. A. and. Pramod A. V., (2011). Security architecture framework and secure routing protocols in wireless sensor networks-survey. *International Journal of Computer Science and Engineering Survey,* 2(4),189-204.

30. Uthra, R. A. and Raja S. (2012). QoS routing in wireless sensor networks: A survey. *ACM Computer Surveys*, 45(9), 1,12.

31. Nagar, S., Rajput, S., Gupta A. K. and. Trivedi, M. C. (2017, February, 9-10). *Secure routing against DDoS attack in wireless sensor network*. 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 1-6.

32. Almani, I., Al-Kasasbeh, B. and AL-Akhras, M. (2016). WSN-DS: a dataset for ıntrusion detection systems in wireless sensor networks. *Journal of Sensors*, 2016, 1–16.

33. Ifzarne, S., Tabbaa, H., Hafidi, I. and Lamghari, N. (2021). *Anomaly detection using machine learning techniques in wireless sensor networks.* Journal of Physics: Conference Series, 1743 (1), 012-021.

34. Garcia-Font, V., Garrigues C. and. Rifà-Pous, H. (2016). A comparative study of anomaly detection techniques for smart city wireless sensor networks. *Sensors*, 16(6), 868.

35. Alrajeh, N., Khan, S. Lloret, J., and Loo, Jo. (2014). Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting. *Ad-Hoc and Sensor Wireless Networks*, 22, 109-133.

36. Otoum, S., Kantarci B. and Mouftah, H. (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, 1(2), 68-71.

37. Alshinina, R. and Elleithy, K. (2018). A highly accurate deep learning based approach for developing wireless sensor network middleware. *IEEE Access*, 6, 29885-29898.

38. Pawar, M. and Anuradha, J. (2021). Detection and prevention of black-hole and wormhole attacks in wireless sensor network using optimized LSTM. *International Journal of Pervasive Computing and Communications,* 88-95.

39. Bahsi, H., Nomm, S., La Torre, F. B. (2018). *Dimensionality reduction for machine learning based ıot botnet detection.* 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapur, 1857-1862.

40. Sokolov, S. A., Iliev, T. B., Stoyanov, I. S. (2019). *Analysis of cybersecurity threats in cloud applications using deep learning techniques*. 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Astana, 441-446.

41. Nomm, S., Bahsi, H. (2018). *Unsupervised anomaly based botnet detection in ıot networks*. 17th IEEE International Conference on Machine Learning and Applications (ICMLA*)*, Orlando, 1048-1053.

42. Anthi, E., Williams, L., Słowińska M, Theodorakopoulos, G., and Burnap, P. (2019). A supervised ıntrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal,* 6(5), 9042-9053.

44. Kumar, A., Lim, T. J. (2019). *EDIMA: Early detection of IoT malware network activity using machine learning techniques*. IEEE 5th World Forum on Internet of Things (WF-IoT), Singapur, 289-294.

45. Possebon, I. P., Silva, A. S., Granville, L. Z., Schaeffer-Filho A. and Marnerides, A. (2019). *Improved Network Traffic Classification Using Ensemble.* IEEE Symposium on Computers and Communications *(ISCC),*1-6.

46. Meidan, Y., Bohadana, M. et. al, (2017). Profiliot: a machine learning approach for IoT device identification based on network traffic analysis, *Symposium on Applied Computing (SAC)*, 506–509.

47. Bai, H., Zhang, X. and Liu, F. (2020). Intrusion detection algorithm based on change rates of multiple attributes for WSN, *Wireless Communications and Mobile Computing*, 1-16.

48. Karanja, E., S. Masupe, S., M. Jeffrey, M. (2020). Analysis of Internet of things malware using ımage texture features and machine learning techniques. *Internet of Things,* 9,100-153.

49. Cvitić, I., Perakovic, D., Periša, M. and Botica, M. (2018). *Smart home IoT traffic characteristics as a basis for DDOS traffic detection*. 3rd EAI International Conference on Management of Manufacturing Systems, Dubrovnik, 112-121.

50. Liu, G., Wang, X., Li, X., Hao, J. and. Feng, Z. (2018). *ESRQ: an efficient secure routing method in wireless sensor networks based on Q-Learning*. 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE*)*. 149–155.

51. Anwer M., Muhammad K Farooq U., Waseemullah F. (2021). Attack detection in IoT using machine learning. *Engineering, Technology & Applied Science Research*,11 (3), 7273-727.

52. Okur, C., Dener, M. (2021). Detection of dos attacks using machine learning methods in wireless sensor networks, *El-Cezerî Journal of Science and Engineering,* 8(3), 1550-1564.

53. Okur, C., Dener, M. (2020). *Detecting IoT botnet attacks using machine learning methods,* 13. International Conference on Information Security and Cryptology, Ankara, 21-25.

54. Okur, C., Orman, A., Dener, M. (2022). *DDOS intrusion detection with machine learning models: N-BaIoT data set*. 4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2022), İstanbul, 772-776.

55. Wazirali, R., Ahmad, R. (2021). Machine Learning Approaches to Detect DoS and Their Effect on WSNs Lifetime. *Computers, Materials & Continua,* 70(3).

56. Branitskiy, A. and Kotenko, I. (2017). Hybridization of computational intelligence methods for attack detection in computer networks, *Journal of Computational Science*, 23, 145-156.

57. Islam M., Fahmin, A., Hossain M., and. Atiquzzaman, M. (2020). Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques. *Wireless Personal Communications,*116(3), 1993-2021.

59. Karuppiah A. B., Dal_ah, J., Yuvashri, K., and Rajaram, S. (2015). An improvised hierarchical black hole detection algorithm in wireless sensor networks. *International Conference on Innovations in Computer Science and Information Technologies*, 1-7.

60. Singh, V. P., Jain, S., Singhai, J. (2010). Hello flood attack and its countermeasures in wireless sensor networks. *International Journal of Computer Sciences*,7(3), 23.

61. Hassanzadeh, A., Stoleru, R. and Chen, J. (2011). *Effcient flooding in wireless sensor networks secured with neighborhood keys.* IEEE 7th International Conference of Wireless Mobile Computer Network Commununications (WiMob*),* 8(5), Texas, 119-126.

62. Chaudhari, H. C. and Kadam, L. U. (2011). Wireless sensor networks: Security, attacks and challenges. *International Journal of Networks*, 1(1), 4-16.

63. Alajmi N. M. and Elleithy, K. M. (2015). Selective forwarding detection (SFD) in wireless sensor networks. *Long Island Systems, Applications, and Technology (LISAT)*, Riyad, 18 (1), 1-5.

64. Okur, C. and Dener, M. (2022). *Deep learning based secure routing protocols in wireless sensor networks*. 2. International Congress of Engineering and Natural Sciences Studies, İstanbul, 65-73.

65. Okur, C., Dener, M. (2021). *Analysis of handwritten characters using deep learning methods*. 6th International Congress of Academic Research, Ankara, 18-35.

66. Wu, P. and Guo, H. (2019). *LuNET: a deep neural network for network intrusion detection*. 2019 IEEE Symposium Series on Computational Intelligence, Sidney, 617-624.

67. Hassan M. M., Gumaei A., Ahmed, A., Alrubaian, M., and Fortino, A. G. (2020) A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Science*, 5, 386-396.

68. Vinayakumar R., Alazab, M., Soman K. P., Poornachandran, P., Al-Nemrat A., Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access,* 7, 41525-41550.

69. Al, S. and Dener, M. (2021). STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Computers & Security*,110, 102435.

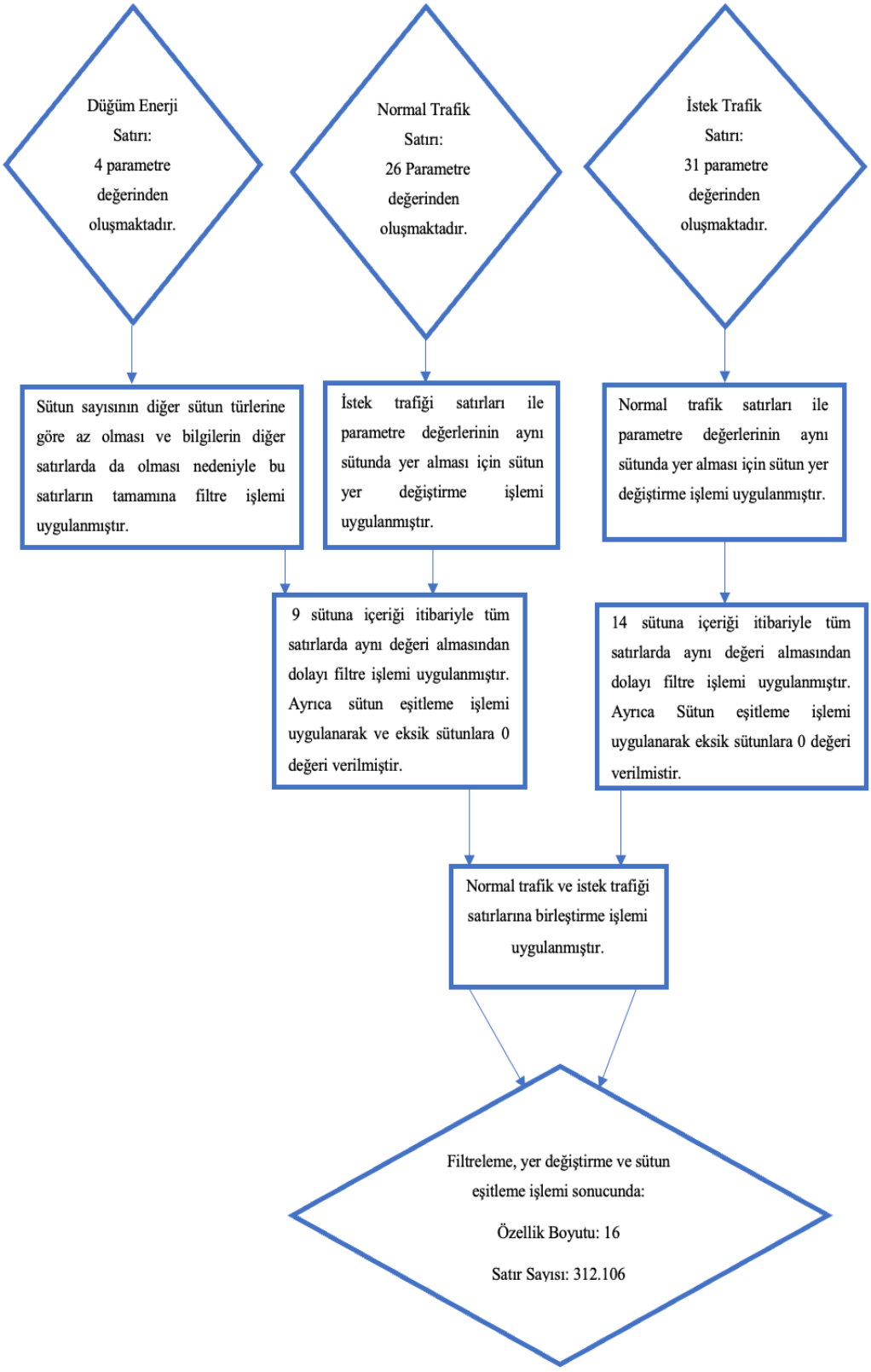
70. Khabir, K., M., Siraj, S. and Habib, A. (2018). *A study on DSR routing protocol in Ad-hoc network for daily activities of elderly living*. 2018 Joint 7th InternationaL Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (ICIVPR), Fukuoka, 573-577.

71. Okur, C., Dener, M. (2020). *Performance comparison of AODV and DSR routing protocols*. V. International Scientific and Vocational Studies Congress Engineering, Ankara, 201-206.

72. Soi V. and Dhaliwal, B.S. (2017). Performance comparison of DSR and AODV Routing Protocol in Mobile Ad hoc Networks. *International Journal of Computational Intelligence Research*, 13(7),1605-1616.

EKLER

EK-1. Veri Seti Elde Etme Aşamaları



EK-2. Veri Dağılımı

Event	Time	S_Node	Node_id	Rest_Energy	Mac_Type_Pckt	Source_IP_Port	Packet_Size	TTL	Hop_Count	Broadcast_ID	Dest_Node_Num	Dest_Seq_Num	Src_Node_ID	Src_Seq_Num	Class
2	276.531081472	65	65	543.083582	800	64.255	48	1	7	9	121	17	0	22	Blackhole
2	276.531081639	66	66	543.035128	800	64.255	48	1	7	9	121	17	0	22	Normal
4	257.449319431	80	80	542.834201	800	42.255	106	6	2	7	121	13	0	18	Forwarding
4	257.449319387	81	81	543.392510	800	42.255	106	6	2	7	121	13	0	18	Normal
2	0.100963687	118	118	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100963745	117	117	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100963833	116	116	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100988333	77	77	599.979723	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100988667	39	39	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
4	102.742493683	85	85	577.444020	800	124.255	106	2	4	4	121	9	0	12	Normal
2	0.100988687	38	38	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100988745	37	37	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.100988833	36	36	599.979722	800	79.255	48	30	1	1	100	0	79	4	Normal
2	0.101383133	1	1	599.979639	800	0.255	48	30	1	1	121	0	0	4	Normal
2	0.101383467	3	3	599.979639	800	0.255	48	30	1	1	121	0	0	4	Normal
2	0.101383667	40	40	599.979638	800	0.255	48	30	1	1	121	0	0	4	Normal
2	0.101383731	42	42	599.979638	800	0.255	48	30	1	1	121	0	0	4	Normal
2	73.1177335572	128	128	584.367970	800	126.255	48	25	6	2	37	0	18	6	Flooding
2	73.1177337738	129	129	584.765362	800	126.255	48	25	6	2	37	0	18	6	Flooding
2	73.1177339905	166	166	584.777576	800	126.255	48	25	6	2	37	0	18	6	Flooding
2	73.1375399928	140	140	585.148454	800	99.255	48	28	3	2	37	0	18	6	Blackhole
2	0.100963833	74	74	599.979722	800	79.255	48	30	1	1	100	0	79	4	Blackhole
2	73.1375400074	62	62	585.064487	800	99.255	48	28	3	2	37	0	18	6	Blackhole
2	73.1338342962	82	82	584.055018	800	122.255	48	24	7	2	37	0	18	6	Forwarding
2	73.1338343026	80	80	584.081040	800	122.255	48	24	7	2	37	0	18	6	Forwarding
2	73.1244471764	83	83	583.703275	800	121.255	48	24	7	2	37	0	18	6	Forwarding
2	0.101383814	43	43	599.979638	800	0.255	48	30	1	1	121	0	0	4	Forwarding

GAZİ GELECEKTİR...