DEVELOPMENT OF A NEW MACHINE LEARNING-BASED METHOD TO DETECT THREATS AND PREVENT ATTACKS ON THE INTERNET OF THINGS AND SIMULATION WITH COOJA

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELİ UNIVERSITY
BY

ALI HAMID AHMED SALEH FAREA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

KOCAELİ 2022

DEVELOPMENT OF A NEW MACHINE LEARNING-BASED METHOD TO DETECT THREATS AND PREVENT ATTACKS ON THE INTERNET OF THINGS AND SIMULATION WITH COOJA

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELİ UNIVERSITY

BY

ALI HAMID AHMED SALEH FAREA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

Prof.Dr. Kerem KÜÇÜK
Supervisor, Kocaeli University .....

Asst.Prof.Dr. Alev MUTLU
Jury member, Kocaeli University .....

Asst.Prof.Dr. Muhammad ASSHAD
Jury member, University of West London .....

Thesis Defense Date:02.11.2022

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez /proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

Beyan ederim.

■ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☐ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler ..... tarafından ..... no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

.....

Ali Hamid Ahmed Saleh FAREA

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin / projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.

Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan ***"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.

☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.

■ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

.....

Ali Hamid Ahmed Saleh FAREA

PREFACE AND ACKNOWLEDGMENT

First and foremost, I would want to extend my gratitude and trust to ALLAH ALMIGHTY, who has given me the fortitude, wisdom, knowledge, and ability to continue my studies and the tenacity to successfully complete my master's degree. I think it would not have been feasible without his blessings and his ability.

I would like to express my appreciation to my supervisor, Prof. Dr. Kerem Küçük, who has cheerfully guided me, answered my queries, provided me with materials, checked my examples, assisted me in myriad ways with the writing, helpfully commented on earlier drafts of this work and review papers, and added comments related to this work as well as supported me during research and made me accurate. Without his supportive knowledge and experience, this work could not be achieved.

Also, I am very grateful to my mother, father, brothers, sisters, wife, and friends for their good humor and support throughout the production of this work, and they are praying for my luck and success. We are also very thankful to our professors in the computer engineering department at Kocaeli University. Without them, this work would not have been able to proceed.

November– 2022 ALI HAMID AHMED SALEH FAREA

CONTENTS

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
PREFACE AND ACKNOWLEDGMENT iii
CONTENTS iv
LIST OF FIGURES i
LIST OF TABLES ii
LIST OF SYMBOLS AND ABBREVIATIONS iii
ÖZET v
GENİŞLETİLMİŞ ÖZET vi
ABSTRACT x
1. INTRODUCTION 1
1.1. Internet of Things 1
1.2. IoT Security 3
1.3. Motivation 5
1.4. Problem Statement 5
1.5. The Scope of Thesis 5
1.6. Objectives of the Thesis 6
1.7. Structure of the Thesis 6
2. BACKGROUND 8
2.1.6LOWPAN Protocol Stack 8
2.1.1. 6LoWPAN Protocols Overview 8
2.1.2. Network- IPV6 layer / Routing Protocol 10
2.1.3. Internet Control Message Protocol 11
2.2.Attacks in IoT 11
2.2.1. DoS Attacks 12
2.2.2. BHA Attacks 13
2.2.3. OOA Attacks 13
3. RELATED WORK 15
3.1. Intrusion Detection System (Rule-Based) 16
3.2. Machine Learning Methods (Anomaly-Based) 19
3.3. Thesis Approaches and Techniques 23
4. METHODOLOGY AND EXPERIMENTAL DESIGN 27
4.1. Proposed Model Design 27
4.2. Simulation Phase 29
4.2.1.DoS Attacks 31
4.2.2.Blackhole Attacks 35
4.2.3.ON-OFF Attacks 39
4.3. Dataset Collection and Manipulation 43
4.4. Pre-processing Phase 45
4.5. Machine Learning Based Approaches and Time Complexity. 46
4.6. Proposed ML-Models Deployment 52
5. ATTACKS EXPERIMENTAL ANALYSIS 54
5.1. Attacks Analysis Method 54
5.2. DoS Attack Analysis 56
5.3. BHA Attack Analysis 58
5.4. OOA Attack Analysis 60

6. ANALYTICS AND SIMULATION TOOLS 62
6.1. Contiki OS 62
6.2. Cooja Simulator 62
6.3. 6LoWPAN Analyzer Tool 62
6.4. Wireshark 62
6.5. Power Trace Tool 63
6.6. Azure Machine Learning Studio 63
7. RESULTS 65
7.1. Performance Metrics for Classification Models 66
7.1.1. The Confusion Matrix for Multiclass Random Forest Model 67
7.1.2. The Confusion Matrix for Multiclass Decision Tree Model 70
7.2. Performance Metrics for Regression Models 73
7.2.1. Performance Matrix for Decision Random Forest Model 73
7.2.2. Performance Matrix for Boosted Decision Tree Model 74
8. CONCLUSION 77
REFERENCES 81
APPENDICES 87
PERSONAL PUBLICATIONS AND ARTICLES 93
BIOGRAPHY 94

LIST OF FIGURES

Figure 1.1. The scope of thesis 6
Figure 2.1. Contiki OS architecture 8
Figure 2.2. LoWPAN devices 9
Figure 2.3. 6LoWPAN protocol stack 9
Figure 2.4. RPL mechanism 10
Figure 2.5. Classification of attacks in RPL-6LoWPAN in IoT 12
Figure 3.1. Intrusion detection system techniques 15
Figure 4.1. Proposed model design for detecting the IoT attacks 27
Figure 4.2. Proposed model implementation to detect the IoT attacks 28
Figure 4.3. Procedures the ML-based approaches to detect IoT attacks 29
Figure 4.4. Basic structure and concept of DoS attacks 32
Figure 4.5. DoS scenario 33
Figure 4.6. DoS nodes output 34
Figure 4.7. Power consumption in a DoS attack 35
Figure 4.8. Basic structure and concept of BHA attacks 36
Figure 4.9. BHA scenario 36
Figure 4.10. Effect BHA on topology 37
Figure 4.11. Power consumption in a BHA attack 39
Figure 4.12. Basic structure and concept of OOA attacks 40
Figure 4.13. OOA Scenario 40
Figure 4.14. OOA mote output 42
Figure 4.15. Power consumption in a OOA attack 43
Figure 4.16. KoÜ-6LoWPAN-IoT dataset features 44
Figure 4.17. Snippet for decision tree-based Model 51
Figure 5.1. Attacks analysis method 55
Figure 5.2. Radio messages features captured 56
Figure 5.3. Power features captured. 56
Figure 5.5. DoS PCP.File damaged 58
Figure 7.1. Decision tree-based models 65
Figure 7.2. Snapshot performance matrix for classification models 66
Figure 7.3. Snapshot of performance matrix for regression models 73
Figure 7.4. The overall accuracy of decision tree-based models 75
Figure 7.5. New predictive value of DoS attacks 75
Figure 7.6. New predictive value of BHA 76
Figure 7.7. New predictive value of OOA 76
Figure 7.8. New predictive value of Normal 76

LIST OF TABLES

Table 3.1. Proposed IDS via authors to detect certain attacks 17
Table 3.2. Proposed ML techniques by authors to detect attacks in IoT..... 21
Table 3.3. The gaps from previous studies are covered by the thesis. 23
Table 3.4. Thesis problems simulation and analysis phase 25
Table 3.5. Thesis problems solution techniques phase..... 25
Table 4.1. Network configuration in Cooja..... 30
Table 4.2. Observations and features in each attack..... 45
Table 4.3. The computational time complexity for the top ML algorithms 49
Table 4.4. The characteristics offered via edge, fog, and cloud computing..... 52
Table 5.1. Comparison between the NS and MS nodes in DoS attack 58
Table 5.2. Benchmark for increase and decrease between NS and MS in DoS 58
Table 5.3. Comparison between the NS and MS nodes in BHA attack 59
Table 5.4. Benchmark for increase and decrease between NS and MS in BHA..... 60
Table 5.5. Comparison between the NS and MS nodes in OOA attack..... 61
Table 5.6. Benchmark for increase and decrease between NS and MS in DoS 61
Table 7.1. Multiclass random decision forest confusion matrix overall classes 67
Table 7.2. BHA class confusion matrix in MRF 68
Table 7.3. DoS attack class confusion matrix in MRF..... 68
Table 7.4. Normal class confusion matrix in MRF 69
Table 7.5. OOA class confusion matrix in MRF..... 69
Table 7.6. Multiclass decision tree confusion matrix overall classes..... 70
Table 7.7. BHA class confusion matrix in MDT..... 71
Table 7.8. DoS attack class confusion matrix in MDT 71
Table 7.9. Normal class confusion matrix in MDT..... 71
Table 7.10. OOA class confusion matrix in MDT 72
Table 7.11. Performance matrix for decision forest regression model..... 73
Table 7.12. Performance matrix for boosted decision tree model..... 74

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviations

IoT	: Internet of Things
ML	: Machine Learning
OS	: Operating System
KoÜ	: Kocaeli Üniversitesi
6LoWPAN	: IPv6 over Low-Power Wireless Personal Area Networks
IDS	: Intrusion Detection Systems
AMLS	: Azure Machine Learning Studio
RF	: Random Forest
DT	: Decision Tree
BDT	: Boosted Decision Tree
KNN	: K-Nearest Neighbors
SVM	: Support Vector Machine
ANN	: Artificial Neural Network
CPU	: Central Processing Unit
LLN	: Low-power and Lossy Networks
DoS	: Denial-of-Service Attack
BHA	: Black Hole Attack
OOA	: ON-OFF Attacks
IEEE	: Institute of Electrical and Electronics Engineers
IPv6	: Internet Protocol Version 6
IETF	: Internet Engineering Task Force
URL	: Uniform Resource Locator
RPL	: Routing Protocol for Low-power lossy networks
CoAP	: Constraint Application Protocol
UDP	: User Datagram Protocol
TCP	: Transmission Control Protocol
RFC	: Request for Comments
DAG	: Directed Acyclic Graph
DODAG	: Destination Oriented DAG
DIS	: DODAG Information Solicitation
DIO	: DODAG Information Object
DAO	: Destination Advertisement Object
ACK	: Acknowledgement
ICMPv6	: Internet Control Message Protocol
WSNs	: Wireless Sensor Networks
LR	: logistic Regression
NB	: Naive Bayes
MAC	: Medium Access Control
ID	: Identity
HBH	: Hop-By-Hop
RDC	: Radio Duty Cycling
LPM	: Low Power Mode
NS	: Normal Scenario
MS	: Malicious Scenario

MAE	: Mean Absolute Error
RMSE	: Root Mean Square Error
RSE	: Relative Squared Error
CD	: Coefficient of Determination
MRF	: Multiclass Random Forest
MDT	: Multiclass Decision tree
AI	: Artificial Intelligence
6MID	: 6LoWPAN Mircochain based Intrusion Detection
EU FP7	: Seventh Framework Programme of the European Union
T-SIEWE	: Trust and Strainer-based Intrusion Detection
AODV	: Ad hoc On-Demand Distance Vector Routing
ANNMLP	: Artificial Neural Network Multilayer Perceptron
MQTT	: Message Queue Telemetry Transport

NESNELERİN İNTERNETİNDE TEHDİTLERİ ALGILAMAK VE SALDIRILARI ÖNLEMEK İÇİN MAKİNE ÖĞRENMESİ TABANLI YENİ BİR YÖNTEM GELİŞTİRİLMESİ VE COOJA İLE BENZETİMİ

ÖZET

Nesnelerin İnterneti (IoT), 20.yüzyılın en umut verici teknolojilerinden biridir. Bu teknoloji o kadar avantajlı ve uygun özelliklere sahip ki, günlük uygulamalarımızda bu teknolojik çözüme tamamen bağımlı hale gelinmiştir. Var olan avantajlarının yanında, IoT ekosistemleri çeşitli saldırılara karşı hassas olduğu bilinmektedir. Donanım "akıllı" hale geldiğinde, tehditlere karşı savunmasızlığında artış olmaktadır. Bu nedenle, IoT güvenliğinin, heterojenlik ve birçok kısıtlamaya sahip dinamik bir ekosistem nedeniyle zor olduğu öngörülmektedir. Bu tezde, Makine Öğrenimi (Machine Learning, ML) tabanlı yaklaşımları kullanarak IoT'deki çeşitli saldırıları tespit etmek ve önlemek için bir yaklaşım önerilmektedir. Bu nedenle yeni bir veri seti (KoÜ-6LoWPAN-IoT veri seti) oluşturmak için Contiki İşletim Sistemi (OS) üzerinden IoT'de saldırılarından olan üç saldırı tipi ele alınmıştır. Bu veri kümesi, kötü niyetli düğümlerden ve normal düğümlerden gelen verilerin bir karışımını içermektedir. Ayrıca, bu veri seti, IoT saldırılarını tespitinde nihai bir karar vermek için karar ağacına göz önüne alan ML tabanlı yaklaşımlarında kullanılabilmektedir. Ayrıca, normal ve kötü niyetli senaryoları ayırt etmek için saldırılar analiz edilmiştir. Analizler, Saldırı Tespit sistemleri (IDS) için bir girdi olarak kullanılabilecek olmasının yanı sıra kötü niyetli faaliyetlerden etkilenen değerlendirmelerin parametrelerini ve özelliklerini anlamak için de kullanılabilmektedir. Ayrıca makine öğrenimi modellerinde kullanılabilecek veri kümelerinin özelliklerindeki i. anormal davranışların tespitinde de faydalı olabilmektedir. Ayrıca, IoT 'deki saldırıları tespit etmek için kullanılacak en hafif ve en uygun modelleri bulmak için en iyi ML algoritmalarının zaman karmaşıklığını araştırılmıştır. Buna ek olarak, modellerin hafifliği ve doğruluğuna dayalı olarak model dağıtımını önerilmiştir. Karar ağacı tabanlı yaklaşımlar, Cooja simülatörü aracılığıyla oluşturulan KoÜ-6LoWPAN-IoT veri setini verimli bir şekilde manipüle etmek, anormal davranışları tespit etmek ve kötü niyetli faaliyetleri sınıflandırmak gibi yeteneklere sahiptir. Karar ağaca yapısı aynı zamanda en hafif ve en iyi modeli elde etmek için kullanılmıştır. Bu tez kapsamında, modellerde IoT özelliklerine dayalı veri seti ile denemeler yapmak ve sonuç almak için ML çerçevesi sağlayan Azure Microsoft Makine Öğrenimi aracı (Azure Machine Learning Studio, AMLS) kullanılmıştır. Ayrıca, tahminlerin sonuçlarını elde etmek için yeni örnekler ve harici bir kaynak aracı kullanarak modeller doğrulanmıştır. Sonuç olarak, çok sınıflı rastgele orman (RF) karar tabanlı yaklaşım, KoÜ-6LoWPAN-IoT veri kümesi için IoT saldırılarını tespit etmede %98,9 genel doğruluk elde ederken karar ağacı (DT), karar ormanı ağacı regresyonu ve güçlendirilmiş karar ağacı sırasıyla %87,7, %93,2 ve %87,1'e ulaşan regresyon sonuçları tespit edilmiştir.

Anahtar Kelimeler: Cooja Simülatör Tabanlı, IoT Güvenliği, Karar Ağacı Tabanlı Modeller, ML Tabanlı Yaklaşımlar, Saldırılar Analizi**.**

NESNELERİN İNTERNETİNDE TEHDİTLERİ ALGILAMAK VE SALDIRILARI ÖNLEMEK İÇİN MAKİNE ÖĞRENMESİ TABANLI YENİ BİR YÖNTEM GELİŞTİRİLMESİ VE COOJA İLE BENZETİMİ

GENİŞLETİLMİŞ ÖZET

Nesnelerin İnterneti (IoT), 20.yüzyılın en umut verici teknolojilerinden biridir. Bu teknoloji o kadar avantajlı ve uygun özelliklere sahip ki, günlük uygulamalarımızda bu teknolojik çözüme tamamen bağımlı hale gelinmiştir. Var olan avantajlarının yanında, IoT ekosistemleri çeşitli saldırılara karşı hassas olduğu bilinmektedir. Donanım "akıllı" hale geldiğinde, tehditlere karşı savunmasızlığında artış olmaktadır. Bu nedenle, IoT güvenliğinin, heterojenlik ve birçok kısıtlamaya sahip dinamik bir ekosistem nedeniyle zor olduğu öngörülmektedir. Bu tezde, Makine Öğrenimi (Machine Learning, ML) tabanlı yaklaşımları kullanarak IoT'deki çeşitli saldırıları tespit etmek ve önlemek için bir yaklaşım önerilmektedir. Bu nedenle yeni bir veri seti (KoÜ-6LoWPAN-IoT veri seti) oluşturmak için Contiki İşletim Sistemi (OS) üzerinden IoT'de saldırılarından olan üç saldırı tipi ele alınmıştır. Bu veri kümesi, kötü niyetli düğümlerden ve normal düğümlerden gelen verilerin bir karışımını içermektedir. Ayrıca, bu veri seti, IoT saldırılarını tespitinde nihai bir karar vermek için karar ağacına göz önüne alan ML tabanlı yaklaşımlarında kullanılabilmektedir. Ayrıca, normal ve kötü niyetli senaryoları ayırt etmek için saldırılar analiz edilmiştir. Analizler, Saldırı Tespit sistemleri (IDS) için bir girdi olarak kullanılabilecek olmasının yanı sıra kötü niyetli faaliyetlerden etkilenen değerlendirmelerin parametrelerini ve özelliklerini anlamak için de kullanılabilmektedir. Ayrıca makine öğrenimi modellerinde kullanılabilecek veri kümelerinin özelliklerindeki i. anormal davranışların tespitinde de faydalı olabilmektedir. Ayrıca, IoT 'deki saldırıları tespit etmek için kullanılacak en hafif ve en uygun modelleri bulmak için en iyi ML algoritmalarının zaman karmaşıklığını araştırılmıştır. Buna ek olarak, modellerin hafifliği ve doğruluğuna dayalı olarak model dağıtımını önerilmiştir. Karar ağacı tabanlı yaklaşımlar, Cooja simülatörü aracılığıyla oluşturulan KoÜ-6LoWPAN-IoT veri setini verimli bir şekilde manipüle etmek, anormal davranışları tespit etmek ve kötü niyetli faaliyetleri sınıflandırmak gibi yeteneklere sahiptir. Karar ağaca yapısı aynı zamanda en hafif ve en iyi modeli elde etmek için kullanılmıştır. Bu tez kapsamında, modellerde IoT özelliklerine dayalı veri seti ile denemeler yapmak ve sonuç almak için ML çerçevesi sağlayan Azure Microsoft Makine Öğrenimi aracı (Azure Machine Learning Studio, AMLS) kullanılmıştır. Ayrıca, tahminlerin sonuçlarını elde etmek için yeni örnekler ve harici bir kaynak aracı kullanarak modeller doğrulanmıştır. Sonuç olarak, çok sınıflı rastgele orman (RF) karar tabanlı yaklaşım, KoÜ-6LoWPAN-IoT veri kümesi için IoT saldırılarını tespit etmede %98,9 genel doğruluk elde ederken karar ağacı (DT), karar ormanı ağacı regresyonu ve güçlendirilmiş karar ağacı sırasıyla %87,7, %93,2 ve %87,1'e ulaşan regresyon sonuçları tespit edilmiştir.

Nesnelerin İnterneti, bilgi toplayan, aktaran ve çevreleriyle etkileşime giren sensörler, aktüatörler, mikrodenetleyiciler ve akıllı cihazlar içeren fiziksel nesnelerden oluşan bir ağdır ve bu cihazların minimum insan müdahalesi ile veri oluşturmasına ve veri alışverişinde bulunmasına olanak tanır. Gerçek ortam verileri sensörler tarafından toplanır, mikrodenetleyici tarafından işlenir ve daha sonra olaylar oluşturmak için aktüatörler tarafından harekete geçirilir. Bu yapılar kendi kendini yapılandırır ve uzaktan kontrol edilebilir. Sensörler aracılığıyla veri toplama ve bulutta depolama kapasitesiyle IoT, akıllı evler ve akıllı şehirler oluşturmak için fiili standart haline gelmiştir. IoT en

umut verici teknolojilerden biridir ve dünya şimdiden çeşitli IoT teknolojilerini kullanmaya başlamıştır. Akıllı şehirler, bina otomasyonu, güvenlik, gözetleme sistemleri, lojistik, sağlık, ekonomi, afet ve askeri sistemler gibi çok çeşitli uygulamalarla etkileşim halinde olduğu gibi çeşitli protokoller aracılığıyla birbirleriyle haberleşir. Bu nedenle, günlük uygulamalarımızda en iyi eforla ve gerçek zamanlı ile güvenmemizi sağlayan çok sayıda özellik sunar. IoT bulutu, büyük miktarda verinin toplanması, işlenmesi, yönetilmesi ve depolanması için yetenekler sağlar. Bu verilere, endüstriler, otoriteler, izleme araçları ve ilgili hizmetler aracılığıyla uzaktan kolayca erişilebilir ve gerektiğinde karar vermelerine olanak tanır.

Bugün var olan bir teknik, bulut bilişimi IoT ile bütünleştirir. Bulut depolama, nesneler ve uygulamalar arasında ince bir ara katman görevi görür. Büyük Veri olarak da bilinen IoT cihazları tarafından üretilen büyük miktarda veri, bağlı cihazlar arasındaki etkileşimin bir ürünüdür ve internet altyapısına önemli miktarda talep getirir. Bu nedenle, kuruluşlar bu yükü azaltmak için bir çözüm veya değiştirme arayışına girmeye zorlandı ve isteğe bağlı ve sınırsız depolama kapasitesi gibi sanal hizmetler sunarak bu sorunu çözmek için bulut bilişim tanıtıldı. Bu iki teknoloji birbirini tamamlar ve günlük görevleri daha etkili hale getirmek için birlikte çalışır. Ancak hizmet sağlayıcıya güvenilmediği sürece bulut bilgi işlem güvenliği ve gizlilik endişeleri çok yüksektir. Bu nedenle bulut bilişim risklidir ve gizlilik çok düşüktür. Sis bilişim, bulut ile uç arasında, verileri yerel olarak işlemeye veya buluta göndermeye karar verdiği bir ara katman olsa da, verilerin gizliliği orta düzeydedir. Uç bilgi işlem, verileri işler ve uçta gerçek zamanlı olarak kararlar verir, bu da çok yüksek gizlilik sağlar.

IoT teknolojilerinin heterojenlik, otomatik yapılandırma, dinamik ekosistem, akıllılık, büyük ölçek ve bağlanabilirlik olarak tanımlanan belirli ortak özellikleri vardır. Örneğin IoT ekosistemi, son derece farklı teknolojiler ve protokoller, uyarlanabilir protokoller, çevresel değişikliklere uyum sağlamak için etkilenebilecek çeşitli faktörler vb. içerir. Bu bileşenler (büyük ölçekli), paylaşmak için işbirliğine dayalı ve akıllı bir şekilde birlikte çalışır. Çoğu durumda, başvuru sahibine güvenli ve güvenilir hizmetler sunmak için bağlı cihazlar gerekir. Teknolojilerin her geçen gün gelişmesi, IoT ekosisteminin çeşitli özelliklerini ve tekniklerini artırmakta, dolayısıyla yeni güvenlik endişelerinin yanı sıra tam olarak ele alınamayan güvenlik açıklarını gündeme getirmektedir.

Günümüzde, çeşitli kaynaklara sahip çeşitli boyutlardaki cihazlar birbirine bağlıdır. Nesnelerin İnterneti, kullanışlılıklarını artırmak için İnternet'e bağlanmayı amaçlayan bir ana plandır. Düşük Güçlü Kablosuz Kişisel Alan Ağları (LoWPAN) için IEEE 802.15.4 protokolünü, çok büyük bir adres alanına sahip olan ve bir milyar cihazın internete bağlanmasına izin verecek olan IPv6 ağ protokolü ile entegre etmenin bir yolunu bulmak gerekiyordu. 6LoWPAN teknolojisi ile, bu soruna uygun bir yanıt, IoT konseptinin gerçeğe dönüşmesine ve küçük nesnelerin veya kısıtlı cihazların (sınırlı CPU, bellek) , ve güç) internete bağlanması sağlanmıştır. İnternet'e bağlı, kaynakları kısıtlı çok sayıda cihaza yayılarak İnternet'in kapsamını ve erişilebilirliğini önemli ölçüde artırırken güvenlik tehditlerine karşı savunmasızlığını artırabilir. Bu nedenle 6LoWPAN, güvenlik gibi bir dizi sorunun ve sorunun yalnızca başlangıcıydı. 6LoWPAN ağları, güvenlik önlemleri sağlayamaması nedeniyle düğüm kaynaklarını tüketen ve ağa zarar veren bir dizi saldırıya açıktır. IoT, akıllı uzaktan yönetim yoluyla verimliliği ve üretkenliği artırabilir, ancak aynı zamanda saldırı potansiyelini de artırır. Son zamanlarda, IoT sistemlerine yönelik risk ve potansiyel tehlikeleri en aza indirme gerekliliği akademik ilgi çekmektedir.

Günümüzde IoT, tehditlerde ve güvenlik açıklarında bir artışla karşı karşıyadır. Tipik olarak, geleneksel ağlarda homojen cihazlar arasında iletişimimiz var, ancak IoT sistemlerinde, heterojen cihazlar ve farklı protokoller üzerinde çalışan ağlar arasında etkileşim vardır. Ayrıca IoT, sınırlı CPU işleme, küçük bellek ve düşük güç tüketimi olan son derece kısıtlı düğümlerden oluşan bir ağdır. Düşük Güçlü Kablosuz Kişisel Alan Ağları üzerinden IPv6 (6LoWPAN), IEEE 802.15.4 ağları üzerinden IP ileten ve IEEE 802.15.4 standardını karşılayan nesnelerden oluşan özel bir düşük güçlü ve kayıplı ağ (LLN) biçimidir. Sonuç olarak, 6LoWPAN cihazı, IoT ekosisteminde güvenliği artırmak için araştırmacıların dikkat etmesi gereken çok önemli bir konudur. Bu nedenle, 6LoWPAN cihazları için akıllı bir güvenlik planı geliştirmek, her türlü riski değerlendirmemizi ve IoT ekosistemini daha güvenli ve güvenilir hale getirmemizi sağlayacaktır. Mevcut güvenlik teknikleri, belirli IoT saldırılarına karşı savunma yapmak için kullanılabilir. Bununla birlikte, geleneksel yaklaşımlar, teknolojik gelişmeler ve çeşitli saldırı türleri ve şiddet seviyeleri karşısında yetersiz kalabilir. Bu nedenle, birçok açıdan işbirliğini geliştirmek için IoT ve Makine Öğrenimi (ML) teknolojilerini birbirine bağlamak temel ve önemlidir. Bu nedenle, IoT cihazlarının/nesnelerinin ve önceki bilgi ve deneyimlere dayalı sistemlerin davranışlarını öğrenmek ve analiz etmek için IoT'de makine öğreniminin etkinleştirilmesi, IoT ekosisteminin sıklıkla anormal koşulların neden olduğu beklenmedik bozulmayı etkili bir şekilde yönetmesine olanak sağlayabilir. Bu nedenle makine öğrenimi yöntemleri, çeşitli bilimlerde mevcut ve gelecekteki sorunları çözmek için çok sayıda yeni araştırma yönünü açan önemli teknik gelişmeler gördü. Bu nedenle, araştırmacılar, hala araştırmacılara açık olan ve daha fazla çalışma gerektiren bir alan olan akıllı yollar bulmaya ilgi duymuşlardır. Yeni nesil IoT sistemleri için güçlü, dinamik olarak geliştirilmiş ve güncel bir güvenlik çözümü gereklidir. Bu nedenle, akıllı ortamları (IoT) daha güvenli ve güvenilir hale getiren güvenlik çözümleri bulmak için akıllı teknolojileri (ML) kullanmak. IoT'nin hızlı büyümesi, onları birçok sorun ve tehdide maruz bırakıyor. Makine öğrenimi yaklaşımları, tutarsız, anormal ve zararlı eylemleri tespit edip sınıflandırmak ve hatalardan kaynaklanabilecek hatalı IoT cihazlarını tespit etmek için güçlü bir teknik olarak kullanılıyor.

Benzer özellikleri ayırt etmek ve anormallikleri tespit etmek, nesnelerin özellikleri ve parametreleri/özellikleri aracılığıyla yapılır. IoT uygulamaları için özel olarak tasarlanmış Etkili Saldırı Tespit Sistemleri (IDS'ler) oluşturulmalıdır. Bu tür IDS'leri eğitmek ve değerlendirmek için güncel ve temsili bir IoT veri kümesine ihtiyaç vardır. Ancak, IDS özellikli IoT sistemlerini ve makine öğrenimi modellerini değerlendirmek için yeterli sayıda karşılaştırmalı IoT veri kümesi bulunmamaktadır. Bunun arkasındaki sebep, IoT ekosistemine yönelik saldırıların analiz eksikliğidir. Saldırı analizi, saldırıları tespit etmek için bir kriter çıkarmayı ve bir saldırı için değil tüm saldırılar için kriterleri karşılayan kriter tabanlı bir veri seti oluşturmayı kolaylaştırır. Ayrıca saldırıların analizi, geliştiricilerin IDS tabanlı IoT özellikleri tasarlamasına ve karşılaştırmalı IoT veri kümeleri üretmesine yardımcı olur. Ayrıca IoT ortamındaki saldırıların özelliklerini/parametrelerini anlamak ve analiz etmek, bir saldırı gerçekleştiğinde etkilenen parametreleri derinlemesine anlamamızı sağlar. Sonuç olarak, saldırgan düğümler, her operasyonda matris değişkenleri değiştiği ve güncellendiği sürece ağ protokolü parametrelerini bir şekilde etkileyebilir ve saldırıları tespit etmek için en uygun yaklaşımı belirlemek için öncü bir yöntem sağlayabilir. Saldırıların analizi, saldırıları değerlendirmek için adil kriterlerin çıkarılmasına yol açar. Bu kriterler, saldırıları tespit etmek için giriş kriteri olarak IDS ve ML tekniklerinde kullanılan özellikler veya özellikler olabilir.

Bu tezde, IoT tabanlı özelliklerden oluşan yeni bir veri kümesi oluşturmak için Contiki OS'yi kullanarak IoT'de örnek olarak üç saldırı gerçekleştirilmiştir. Ayrıca Saldırı Tespit sistemleri (IDS) için bariz bir kriter ve girdi olan saldırılardan etkilenen özellikleri veya Makine Öğrenimi modellerinde (ML) kullanılabilecek veri kümelerinin özelliklerini keşfetmek ve çıkarmak için saldırılar analiz edildi. Ayrıca, IoT saldırılarını tespit etmek için nihai kararı veren makine öğrenimi tabanlı yaklaşımlara dayanan IoT saldırılarını tespit etmek için bir yaklaşım önerdik ve deneyleri yapmak için AMLS'yi kullanılmıştır. Ayrıca, zaman karmaşıklığı ve doğruluğuna dayalı olarak hafif ve optimum makine öğrenimi modellerini ve ayrıca modellerin ağda önerilen dağıtımını araştırılmıştır.

IoT'nin artan kullanımı ve dünyanın tanık olduğu otomasyon ışığında, günlük hayatımızda en iyi çabayla ve kısa sürede problem çözme hizmetleri sağlamaktadır. Belirli bir soruna çözüm bulmak veya geliştirmek için çeşitli disiplinlerden teknolojilerin işbirliği ve etkinleştirilmesi, özellikle kısıtlanan cihazlar ışığında, heterojen bir ekosistemde çalışan bir soruna radikal bir çözüm olabilir. Bu nedenle, IoT ekosistemi için siber güvenlik çözümleri bulmada makine öğrenimi teknolojilerinin etkinleştirilmesi, ilgilenenler için hem zorlayıcı bir problemdir.

Tezin motivasyonu; "Cooja simülatörü tabanlı, makine öğrenimi tabanlı yaklaşımlar, karar ağacı tabanlı modeller, önerilen modellerin IoT ağında konuşlandırılması ve istatistiksel analiz ile hafif ve optimal modelleri kullanarak IoT'de saldırıları tespit etmek ve güvenliği artırmak için önerilen bir yaklaşımı araştırmak" olarak kurgulanmıştır.

Bu tezin ana katkısı şu şekilde özetlenebilir:

- Makine öğrenimi tabanlı yaklaşımlara dayanan IoT saldırılarını tespit etmek için bir yöntem önermek.
- Üç farklı IoT saldırısı uygulamak, Cooja simülatörü tabanlı. Saldırılar, Hizmet Reddi saldırıları (DoS), Kara Delik Saldırısı (BHA) ve ON-OFF Saldırısıdır (OOA).
- IoT özelliklerine dayalı yeni bir veri seti oluşturma (KoÜ-6LoWPAN-IoT veri seti).
- KoÜ-6LoWPAN-IoT veri setinin karar ağacı tabanlı modellere uygulanması ve sonuçların sunulması.
- Normal ve kötü amaçlı senaryoları analiz etme ve IDS ve ML modelleri için kriter girdisi olarak kullanmak üzere her saldırı için değerlendirilen parametreleri keşfetme.
- Zaman karmaşıklığı ve doğruluğuna dayalı olarak IoT saldırılarını tespit etmek için en iyi hafif ve optimum modeli keşfetmek.
- Hafif, ağır ve doğruluk modellerine dayalı olarak modelin IoT ekosistemindeki dağıtımını önermek.

Tezin organizasyonu şu şekilde yapılandırılmıştır: Bölüm 2, IoT ağları için 6LoWPAN protokol yığınına genel bir bakışı açıklar ve simülasyon aşamasında örnek olarak uygulanan üç IoT saldırısının önizlemesini gösterir. Bölüm 3, çalışmamızla ilgili literatür taramasını ve bu tezin katkısını göstermektedir. Bölüm 4 IoT'deki saldırıları tespit etmeye yönelik metodolojiyi göstermenin yanı sıra önerilen yöntemi, simülasyon aşamasını, veri kümesi toplama ve manipülasyonu, karar ağacı tabanlı aşamayı, zaman karmaşıklığını, hafif modelleri, optimal modeli ve önerilen model dağıtımını gösterir. Bölüm 5 Saldırıların analizi ve saldırılardan etkilenen faktörler açıklanmaktadır. Bölüm 6 İşimizi yürütmek için kullandığımız araçları açıklar. Bölüm 7, karar ağacı tabanlı model sonuçlarını tartışır. Bölüm 8, bu çalışmanın sonucunu ve sınırlamaları ve gelecekteki çalışmaları özetler. Ek Bölüm, üç saldırıda kötü amaçlı düğümlerin ve normal düğümlerin kod yapısını göstermektedir.

DEVELOPMENT OF A NEW MACHINE LEARNING-BASED METHOD TO DETECT THREATS AND PREVENT ATTACKS ON THE INTERNET OF THINGS AND SIMULATION WITH COOJA

ABSTRACT

The Internet of Things (IoT) is one of the most promising 3rd-millennium technologies. It has so many attractive and enticing features that we have become completely reliant on this technology in our daily applications. However, the IoT ecosystems are susceptible to a variety of attacks. Once hardware becomes "intelligent", it is vulnerable to threats. Hence, IoT security is considered challenging due to heterogeneity and a dynamic ecosystem with many constraints.

In this thesis, we have proposed an approach to detect and prevent various attacks in the IoT using Machine Learning (ML)-based approaches. Therefore, we developed and implemented three attacks as a sample in the IoT via the Contiki Operating System (OS) to generate a new dataset (KoÜ-6LoWPAN-IoT dataset). This dataset contains a mix of data from malicious nodes and normal nodes. Also, this dataset is used to utilize the ML-based approaches that depend on the decision of the tree to give us a final decision to detect IoT attacks. As well, the attacks were analyzed to distinguish between normal and malicious scenarios. The analyzing is done to understand the parameters and features of evaluations that were affected by malicious activities, which can be used as an input for Intrusion Detection systems (IDS) and features of the datasets that can be employed in Machine Learning (ML) models to detect anomalous behaviors. Furthermore, we investigated the time complexity for the best ML algorithms to find the most lightweight and optimal models to utilize to detect attacks in the IoT. In addition, we proposed model deployment based on the lightness and accuracy of the models.

The decision tree-based approaches have the ability to manipulate the KoÜ-6LoWPAN-IoT dataset efficiently, generated via the Cooja simulator, to detect anomalous behavior and classify malicious activity. It also achieves the lightest and best model. In the context of this work, we utilized Azure Machine Learning Studio (AMLS), which provides the ML framework to experiment with the dataset that is IoT features-based in models and obtain results. Furthermore, we verified the models by using new samples and using an external source to acquire the results of the predictions. As a result, the multiclass random forest (RF) decision-based approach achieved 98.9% overall accuracy in detecting IoT attacks for the KoÜ-6LoWPAN-IoT dataset compared to the decision tree (DT), decision forest tree regression, and boosted decision tree regression, which achieved 87.7%, 93.2%, and 87.1%, respectively.

Keywords: Cooja Simulator-Based, IoT Security, Decision Tree-Based Models, ML-Based Approaches, Attacks Analysis.

1. INTRODUCTION

1.1. Internet of Things

An Internet of Things (IoT) is a network of physical objects containing sensors, actuators, microcontrollers, and smart appliances that gather and transfer information and interact with their surroundings (Laghari et al., 2022) and (Lee et al., 2015) allowing these devices to generate and exchange data with minimal human intervention. Real-world data is collected by the sensors, processed by the microcontroller, and then acted upon by the actuators to produce events. These gadgets are self-configuring and remote-controllable. It will bring about a massive change in the domain of networking. The IoT, with its capacity to gather data via sensors and store it in the cloud, therefore, has become the de facto standard for creating smart systems, smart homes and smart cities. The IoT is one of the most promising technologies and the world is already beginning to utilize various IoT technologies. It communicates with each other via various protocols (Bayılmış et al., 2022) as well as interacts with a wide range of applications, including smart cities, building automation, safety, surveillance systems, logistics, healthcare, economy, calamity and military systems (Pflanzner & Kertesz, 2018) and (Bayılmış et al., 2022). Therefore, it offers a large number of attractive qualities that have made us rely on it in our daily applications with best-effort and real-time (Mekala & Viswanathan, 2017) and (Minh Dang et al., 2019). The IoT cloud provides capabilities for collecting, processing, managing, and storing massive amounts of data. This data may be easily accessed remotely via industries, governments, monitoring tools, and related services, allowing them to make decisions as needed (Ping et al., 2018) and (Johnsen et al., 2018).

A very interesting technique that exists today integrates cloud computing with the IoT (Mekala & Viswanathan, 2017) and (Sikarwar et al. 2020). Cloud storage acts as a subtle intermediate layer between things and apps. The massive amounts of data generated by the IoT devices, also known as Big Data, is a product of interaction between connected devices and puts a significant amount of demand on the internet infrastructure (Mekala & Viswanathan, 2017) and (Minh Dang et al., 2019). Therefore, organizations have been forced to hunt for a resolution or replacement to decrease this load, and cloud computing has been introduced to address this issue by offering on-demand and virtual services like unlimited storage capacity (Abdulkareem et al., 2021). These two technologies

complement one another and work together to make daily tasks more effectively. But cloud computing security and privacy concerns are very high unless the service provider is trusted. Therefore, cloud computing is risky and privacy is very low. While fog computing is an intermediate layer between the cloud and the edge, where it decides to process the data locally or send it to the cloud, the privacy of the data is moderate. Edge computing processes data and makes decisions in real-time at the edge, resulting in very high privacy.

IoT technologies have certain properties in common that are described as heterogeneity, auto-configuring, dynamic ecosystem, smartness, large scale, and connectivity (Pflanzner & Kertesz, 2018) , (Abdulkareem et al., 2021), (Ramakrishnan et al., 2014) and (D. Choudhary, 2019). For example, the IoT ecosystem includes extremely different technologies and protocols, adaptive protocols, a variety of factors that may be influenced in order to adapt to environmental changes, etc. These components (large scale) work together in a cooperative and smart way to share their collected data and services (Hajjaji et al., 2021). In many cases, connected devices are required to offer secure and reliable services to an applicant (Shah & Narmavala, 2018). The development of technologies day by day increases the various characteristics and techniques of the IoT ecosystem, therefore, raising new security concerns (Verma & Ranga, 2020) and (Rghioui et al., 2013) as well as vulnerabilities that cannot be fully addressed by using traditional security techniques.

Nowadays, various sizes of devices with diverse resources are interconnected. The IoT is a master plan that intends to interconnect things to the Internet in order to increase their usefulness (Rghioui et al., 2013). It was necessary to discover a way to integrate the IEEE 802.15.4 protocol for Low Power Wireless Personal Area Networks (LoWPANs) with the IPv6 network protocol, which has a huge address space and will allow a billion devices to connect to the internet. The invention of 6LoWPAN technology was a suitable answer to this problem (Verma & Ranga, 2020) and (Rghioui et al., 2013), allowing the IoT concept to become a reality and allowing the tiny objects or constrained devices (limited CPU, memory, and power) to be connected to the internet. This might expand to a huge number of Internet-connected resource-constrained devices, dramatically enhancing the scope and accessibility of the Internet while increasing its vulnerability to security threats.

Therefore, 6LoWPAN was merely the beginning of a series of problems and issues, such as security (Verma & Ranga, 2020). 6LoWPAN networks are susceptible to a range of attacks that exhaust node resources and damage the network due to its inability to provide security measures (Rghioui et al., 2013) and (Verma & Ranga, 2020). The IoT may improve efficiency and productivity through remote administration that is smart, but it also raises the potential for attacks. Recently, the necessity to minimize risk and potential dangers to IoT systems has attracted academic interest.

1.2. IoT Security

Nowadays, the IoT is facing an increase in threats and security vulnerabilities. Typically, we have communication among homogeneous devices in traditional networks, but with IoT systems, we have interaction between heterogeneous devices and networks that work on different protocols. As well, the IoT is a network made up of extremely restricted nodes which are limited CPU processing, small memory, and low power consumption. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) is a particular form of low-power and lossy network (LLN) that transmits IP over IEEE 802.15.4 networks and is composed of objects that satisfy the IEEE 802.15.4 standard. As a result, 6LoWPAN device considerations it is a very important matter that needs the attention of researchers in order to enhance security in the IoT ecosystem. Therefore, developing a smart security plan for 6LoWPAN devices will allow us to evaluate any risks and make the IoT ecosystem more secure and reliable. Current security techniques may be used to defend against specific IoT attacks. However, the traditional approaches may be inefficient in the face of technological advancements, as well as a variety of attack kinds and severity levels. Thus, it is basic and important to connect IoT and Machine Learning (ML) technologies in order to enhance their cooperation in many aspects. Therefore, enabling ML in IoT for learning and analyzing the behaviors of IoT devices/objects, and systems based on prior information and experiences may allow the IoT ecosystem to effectively manage the unexpected deterioration that is frequently caused by anomalous conditions. Therefore, ML methods have seen significant technical development, opening up numerous new research directions to solve current and future problems in various sciences (Xiao et al., 2018) and (Injadat et al., 2020). Hence, the researchers have taken their interest in finding smart ways, an area that is still open to researchers and that

requires further studies. For the next-generation IoT systems, a powerful, dynamically improved, and up-to-date security solution is necessary. Therefore, utilizing smart technologies (ML) to find security solutions for smart environments (IoT) that make them more secure and reliable. The rapid growth of IoT exposes them to many issues and threats. ML approaches are being used as a strong technique to detect and classify inconsistent, abnormal, and harmful actions and detect incorrect IoT devices that may be due to errors.

Distinguishing between similar things and detecting anomalies is done through the characteristics and parameters/features of things. Effective Intrusion Detection Systems (IDSs) specifically designed for IoT applications must be created. For training and evaluating such IDSs, a current and representative IoT dataset is needed. But there aren't enough benchmark IoT datasets available for evaluating IDS-enabled IoT systems and ML models. The reason behind this is a lack of analysis of attacks on the IoT ecosystem. Attack analysis makes it easy to extract a criterion to detect attacks and create a criteria-based dataset that meets the criteria not for one but for all attacks. As well, an analysis of the attacks helps the developers to design IDS-based IoT features and produce benchmark IoT datasets. Furthermore, understanding and analyzing the features/parameters of attacks in the IoT environment provides us with a deep understanding of the parameters that are affected when an attack occurs. As a result, attacker nodes may affect network protocol parameters in some way as long as the matrix variables change and update in each operation, as well as providing a pioneering method for determining an optimal approach to detecting attacks. Analysis of attacks leads to the extraction of fair criteria for evaluating attacks. These criteria may be characteristics or features that are used in IDS and ML techniques as input criteria to detect attacks.

In this thesis, we implemented three attacks in the IoT as a sample, using Contiki OS to generate a new dataset of IoT-based features. As well, attacks were analyzed to explore and extract the features affected by the attacks, which are an obvious criterion and input for Intrusion Detection systems (IDS) or features of the datasets that can be employed in Machine Learning models (ML). We also proposed an approach for detecting IoT attacks that are based on ML-based approaches that release the final decision to detect IoT attacks and we utilized the AMLS to do the experiments. Furthermore, we explored the

lightweight and optimal ML models based on time complexity and accuracy, as well, as the proposed deployment of models in the network.

1.3. Motivation

In light of the increasing use of the IoT and the automation that the world is witnessing, which provide us with problem-solving services in our daily lives with best-effort and short time. The collaboration and enabling of technologies from various disciplines in finding or enhancing a solution to a specific problem may be a radical solution to a problem that works in a heterogeneous ecosystem, especially in light of the devices being constrained. Therefore, Enabling the ML technologies in finding cyber security solutions for the IoT ecosystem is both challenging and attractive to those interested. Especially if it is in a tangible simulated environment that helps us to implement the attacks and analyze the features and parameters for evaluation and understand them. Therefore, enabling AI to address cyber security in the IoT ecosystem is interesting and attractive to the author.

1.4. Problem Statement

“To investigate a proposed approach to detect attacks and enhance the security in the IoT utilizing Cooja simulator-based, ML-based approaches, decision tree-based models, lightweight and optimal models with proposed models' deployment in IoT network as well as statistical analysis.”

1.5. The Scope of Thesis

Enabling artificial intelligence (AI) to address cyber security in the IoT ecosystem. In a clear sense, employing ML techniques in the IoT ecosystem to detect and analyze attacks and solve security issues in the IoT. Figure 1.1. shows the scope of the thesis.

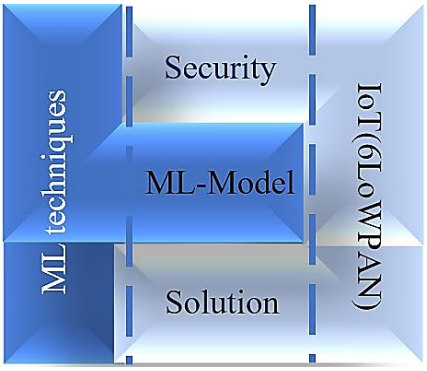


Figure 1.1. The scope of thesis

1.6. Objectives of the Thesis

The main contribution of this thesis is summarized as follows:

- ❖ Proposing a method to detect IoT attacks that relies on ML-based approaches.
- ❖ Implementing three different IoT attacks, Cooja simulator-based. The attacks are Denial-of-Service attacks (DoS), Black Hole Attack (BHA), and ON-OFF Attack (OOA).
- ❖ Generating a new dataset based on IoT features (KoÜ-6LoWPAN-IoT dataset).
- ❖ Applying the KoÜ-6LoWPAN-IoT dataset to decision tree-based models and presenting the results.
- ❖ Analyzing normal and malicious scenarios and exploring the parameters under evaluation for each attack to utilize as criteria input for IDS and ML models.
- ❖ Exploring the best lightweight and optimal model to detect IoT attacks based on time complexity and accuracy.
- ❖ Proposing the model's deployment in the IoT ecosystem based on the light, heavyweight and accuracy models.

1.7. Structure of the Thesis

This thesis is structured as follows:

Chapter 2 Explains an overview of the 6LoWPAN protocol stack for IoT networks and shows a preview of three IoT attacks that were implemented as samples during the simulation phase.

Chapter 3 Illustrates the literature review related to our work and the contribution of this thesis.

Chapter 4 Demonstrate the methodology for detecting attacks in the IoT as well as shows the proposed method, simulation phase, dataset collection and manipulation, decision tree-based phase, time complexity, lightweight models, optimal model, and proposed model deployment.

Chapter 5 Analysis of the attacks and describes the factors affected by the attacks.

Chapter 6 Describes the tools we use to carry out our work.

Chapter 7 Discusses the decision tree-based model results.

Chapter 8 Summarizes the conclusion of this work and limitations as well as future work.

Appendix Section illustrates the code structure of malicious nodes and normal nodes in three attacks.

2. BACKGROUND

2.1. 6LOWPAN Protocol Stack

2.1.1. 6LoWPAN Protocols Overview

IPv6 over low power wireless personal area network (6LoWPAN) is a particular instance of a low power lossy network (LLN) that allows tiny devices with restricted resources to connect to IPv6 networks, and these devices comply with the IEEE 802.15.4 standard (Verma & Ranga, 2020). It supports end-to-end IPv6 connectivity, allowing it to have a direct connection to the Internet with a wide range of networks (heterogeneous devices), including tiny devices (Rghioui et al., 2013) and (Verma & Ranga, 2020). Interoperability is an important consideration when selecting a wireless protocol. Interoperability implies that apps do not need to know the limits of the physical connections that transport their packets. 6LoWPAN devices can communicate with any wireless 802.15.4 devices over any other IP network connection (e.g., Ethernet or Wi-Fi), in contrast to other technologies such as ZigBee devices, which can only communicate with other ZigBee devices (Ismaili et al., 2019). In the network layer, the Internet Engineering Task Force (IETF) proposed routing protocol for low-power lossy networks (RPL) is the most popular routing protocol for LoWPAN devices in the IoT (Jyothisree & Sreekanth, 2019) as well as used in academia and industry. Therefore, Contiki-OS provides work on the IoT and wireless sensor networks that are constrained and operate on a 6LoWPAN protocol stack (Contiki tutorials, 2016).

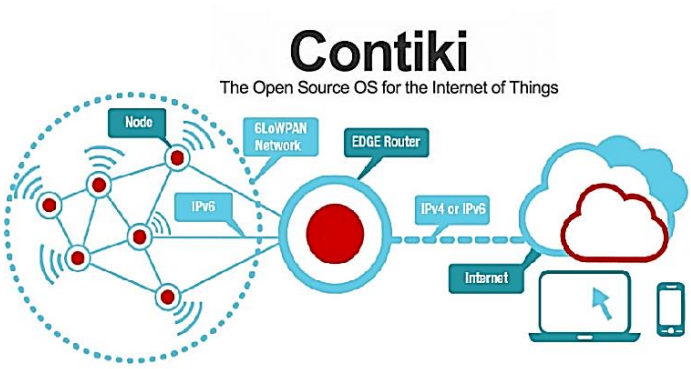


Figure 2.1. Contiki OS architecture (URL1).

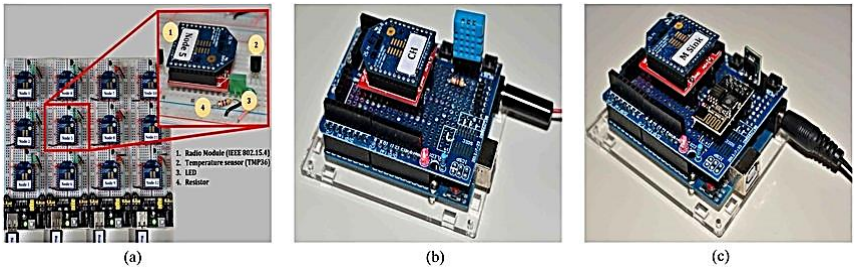


Figure 2.2. LoWPAN devices (Al-Kaseem & Al-Raweshidy, 2017)

Figure 2.1. illustrates the Contiki OS architecture (URL1) while Figure 2.2. shows the LoWPAN devices (Al-Kaseem & Al-Raweshidy, 2017).In comparison to the normal Internet stack, the 6LoWPAN stack contains an extra layer, which is known as the LoWPAN adaption layer. The adaption layer rests above the IEEE 802.15.4 layer, immediately below the network layer. The adaption layer offers header compression, fragmentation, and reassembly, as well as packet forwarding services, allowing IPv6 connections to be provided to extremely tiny devices linked to the internet (Hennebert & Dos Santos, 2014). Therefore, the IPv6 packets are encapsulated to be sent to the underlying link-layer via the adaption layer.

Application Layer Ex. CoAP
Transport Layer UDP/TCP
Network Layer IPV6/ICMP6/RPL
6LoWPAN Adaptation 6LoWPAN
Data Link Layer IEEE 802.15.4 MAC
Physical Layer IEEE 802.15.4 PHY

Figure 2.3. 6LoWPAN protocol stack

Figure 2.3. illustrates the structure of the 6LoWPAN stack and the used protocols. The application layer in the 6LoWPAN Stack contains lightweight protocols such as

Constraint Application Protocol (CoAP) that were created for the IoT and were inspired by HTTP, assuming that UDP may be used without impedance in security (RFC 8323). CoAP over UDP's message layer supports reliable delivery, basic congestion control, and flow control. It was designed with simplicity in mind, with a minimal code footprint and a small, lightweight message size (Hennebert & Dos Santos, 2014). As a recommendation from the developers (IETF), due to the 802.15.4 MAC/PHY frame size limits, UDP is a better fit for the 6LoWPAN stack than standard TCP at the transport layer with a large size header up to 60 bytes (RFC 8323).

2.1.2. Network- IPV6 layer / Routing Protocol

RPL is a routing protocol designed for low-power and lossy networks, and it has become the preferred routing protocol for IoT.

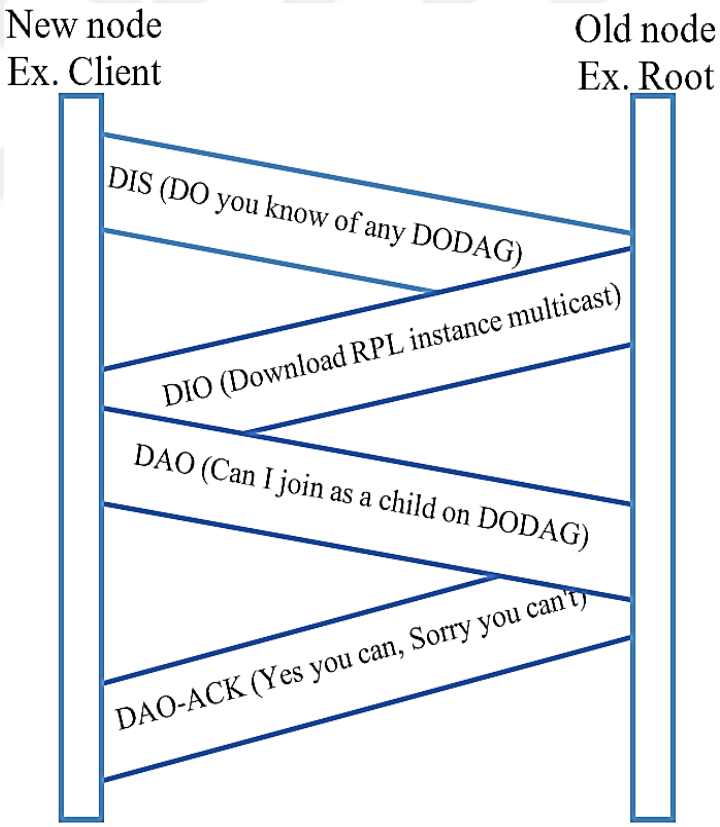


Figure 2.4. RPL mechanism

It is a distance-vector routing protocol that routes the data to a destination (sink) with a short path (Optimal Path). RPL was designed to be highly adaptive to network conditions and to provide alternate routes (Chen et al., 2012). One of the main goals of RPL is to

construct topologies of the network (Charles & Kalavathi, 2018). The resulting routes from a Directed Acyclic Graph (DAG) are the network topology. There is only one Destination Oriented Directed Acyclic Graph (DODAG) per root (sink) and it is the data to a root (Chen et al., 2012). The RPL consists of four control messages that are used in the formation and maintenance of the network topology. Which are as follows: DIS, DIO, DAO, and DAO-ACK are acronyms for DODAG Information Solicitation, DODAG Information Object, Destination Advertisement Object, and Destination Advertisement Object Acknowledgement, respectively. A node can utilize DIS to explore for DODAGs in its general vicinity. The DIO contains data that enables a node to find an RPL instance, understand its configuration parameters, choose a DODAG parent set, and keep the DODAG up to date (Chen et al., 2012).The DODAG in the node uses the DAO to communicate destination information upward. The DAO message is unicast by the child to the specified parent. DAO-ACK is a message sent back to the DAO sender (Charles & Kalavathi, 2018). Figure 2.4. shows the diagram of control messages in the RPL.

2.1.3. Internet Control Message Protocol

Every IPv6 node must successfully implement ICMPv6 since it is an integral part of IPv6. The IPv6 nodes utilize ICMPv6 to report packet processing errors and conduct additional internet-layer operations, including diagnostics, such as ICMPv6 "pinging" and multicast member reporting (RFC 1885).

2.2. Attacks in IoT

6LoWPAN protocol stack is vulnerable to attack due to the IoT devices are connected to an unsecured internet, therefore providing security in the IoT is critical (Seliem & Elgazzar, 2019). An attacker can capture, clone, tamper with, or even destroy LoWPAN devices (Kim et al., 2012). Therefore, 6LoWPAN channels are generally vulnerable to a variety of security risks. The characteristics of 6LoWPAN technologies may provide attractive services compared to their peers (Gracia et al., 2016). However, they may be more vulnerable to attacks due to heterogeneity, dynamic ecosystems, etc. Although the link-layer offers encryption (Kim et al., 2012), it may not be sufficient to guarantee security to both data and signaling packets. It may be encrypted harmful packages without detecting them. The data may be encrypted to maintain confidentiality between the

endpoints (Tanveer et al., 2020). But it is difficult to detect and know whether the data sent is reliable, has not been tampered with, dropped, or has been breached via malicious action. Attacks may occur in different layers with different severity. Figure 2.5. illustrates the attack types and classification in RPL-6LoWPAN in IoT (Verma & Ranga, 2020). In this thesis, we will preview three attacks DoS, BHA, and OOA. which may be exposed to the IoT ecosystem and makes IoT devices at a critical point.

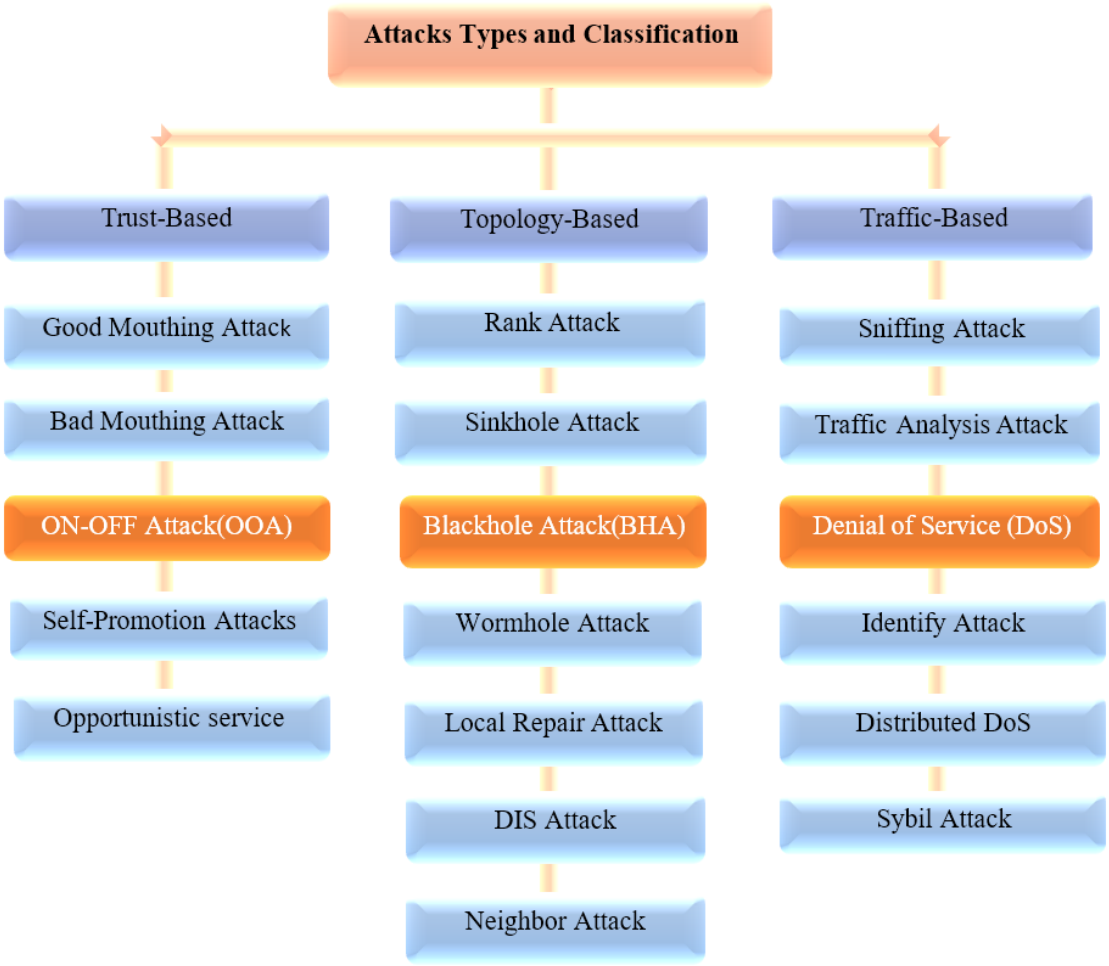


Figure 2.5. Classification of attacks in RPL-6LoWPAN in IoT

2.2.1. DoS Attacks

DoS is one of the most well-known network-attacking behaviors, interrupting and blocking legitimate user requests by flooding the host server with a large number of requests .it is an attempt to prevent the targeted users from accessing resources. This attack may occur in RPL via UDP packet flooding (Ioulianou & Vassilakis, 2020) and

(Abughazaleh et al., 2020). Therefore, the attacker node sends too many requests to the root (sink), preventing normal users from accessing it in their usual way (Dobbins et al., 2021) and (Abughazaleh et al., 2020). The attacking node exploits the target node (destination, sink) by sending too many requests, which hinders normal nodes from reaching the target because the target node is busy processing the flood of requests from the malicious node (Sicari et al., 2018). The DoS attack intends to bring shut down a device or network such that its intended users are unable to access it (Djuitcheu et al., 2022). DoS attacks do this by sending information that causes a breakdown or by overloading the target with traffic. In both periods, the DoS attack hinders normal nodes (legitimate nodes) of the resources or service they were expecting (Sicari et al., 2018) and (Djuitcheu et al., 2022).

2.2.2. BHA Attacks

BHA, which is one of the most dangerous attacks in RPL in which the malicious node drops the packets that are received from its neighbors to forward to their destination (Neerugatti & Reddy, 2019) and (Ahmed & Ko, 2016), may drop all the packets, which is called a complete black hole, or drop some packets, called a selective forwarding attack, and this is cleverer because it is not observed and the network topology is not affected (Ali et al., 2018) and (Ahmed & Ko, 2016). Black-hole attacks initiate when a node discards all packages it is supposed to forward to their destination. Periodically, the malicious node is set up incorrectly to drop the packets (Kale & Bhosale, 2021). The malicious node drops or discards all the packages from neighboring nodes and does not forward them to their destination (Neerugatti & Reddy, 2019), (Ahmed & Ko, 2016), (Ali et al., 2018) and (Kale & Bhosale, 2021). The black hole is eloquent when the malicious node as intermediate between two groups of nodes and these groups can't communicate just through the malicious node which may lead to separated many nodes (Tanwar, 2013).

2.2.3. OOA Attacks

Trustworthiness in IoT devices is critical. OOA is one type of attack that affects trust in the IoT and the devices and objects don't trust each other (Farea & Küçük, 2021) , (Mendoza & Kleinschmidt, 2015) and (Nasution et al., 2020). The on-off attacks are difficult attacks to detect behavior it due to inconsistent behavior manner (Labraoui &

Sekhri, 2018) and (Perrone & Nelson, 2006). OOA is a kind of selective attack (inconsistent behavior) in which the malicious node switches from malicious to normal and back again to avoid being classified as a low-trust node, allowing it to remain undiscovered while inflicting harm and making the nodes suspicious to their neighbors (Mendoza & Kleinschmidt, 2015) and (Nasution et al., 2020). The malicious node initiates and launches the attacks when the status of the device is active (ON status) and transitions to a normal mode in the same status. This leads to difficulty detecting the behavior due to changes (inconsistent behavior) while the malicious node is an inactive node when the status is passive (off status) without doing anything (Perrone & Nelson, 2006), (Moradi et al., 2019) and (Suryani & Widyawan, 2018). The malicious node alternately launches the malicious packages and normal packages if the status is active and does nothing if the status is inactive (Perrone & Nelson, 2006). An OOA transmits bad and good data in an alternate manner (Labraoui & Sekhri, 2018) and (Perrone & Nelson, 2006).

3. RELATED WORK

There have been a variety of related studies on the topic of IoT security. Researchers are still working in this field to find intrusion detection systems (IDS). An IDS is a technique for detecting and evaluating malicious activities in the network. Therefore, the literature review in this thesis is directed in two directions related to IoT security, which are intrusion detection systems (Rule-based) and machine learning methods (Anomaly-based) that are used to address IoT security issues. Hybrid-based IDS integrates Rule-based and Anomaly-based. Figure 3.1. shows the intrusion detection system techniques.

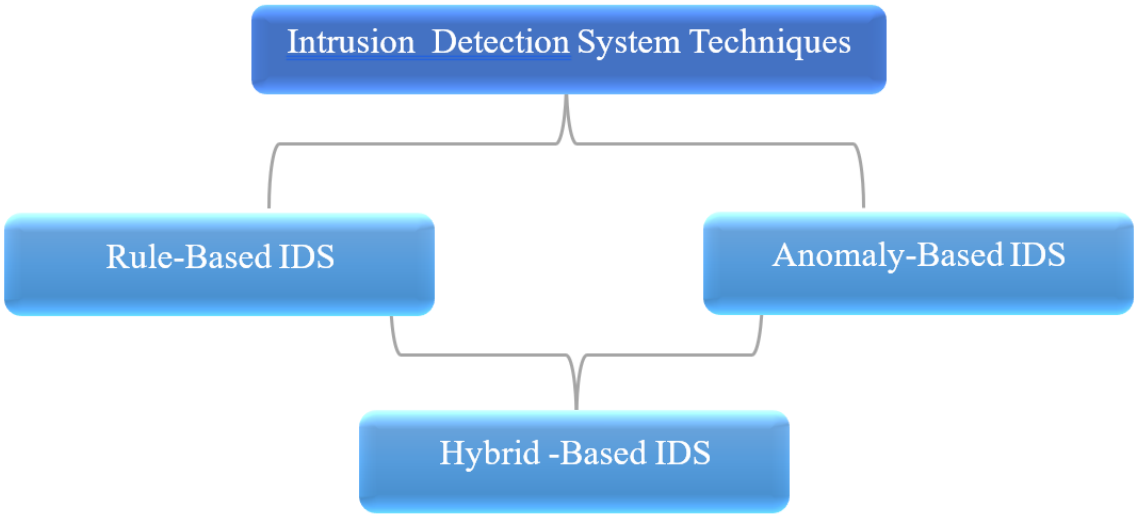


Figure 3.1. Intrusion detection system techniques

Rule-based IDS If a rule is matched in the rule-database, it can detect attacks; otherwise, it cannot detect them. The rules are identified by humans manually based on the features of specific attacks. A framework may be built from scratch depending on rules in the database and constraints, so it is costly. It can detect known attacks (rule-defined), but it is not adaptive and can't renewably detect unknown attacks (Choudhary & Kesswani, 2019) and (Zhang et al., 2016). Anomaly-based IDS is a technique to trace unknown behavior or malicious activities and alert them to classify them as anomalous. It employs machine learning algorithms that depend on the datasets, so it isn't costly. The model trains and learns automatically. It's adaptive and can renewably detect unknown attacks (Jose et al., 2018) and (Zhang et al., 2016). Therefore, this thesis depends on machine learning methods (Anomaly-based) and Rules-Based. Firstly, we used anomaly-based and got the score of labels and probabilities. The rules-based was used after getting the

results from anomaly-based to adaptive with the new label unknown. The third part of this chapter highlights the aspects of the thesis and its differences from previous studies.

3.1. Intrusion Detection System (Rule-Based)

Existing research in the literature for IoT security areas offers numerous security approaches. Many approaches have been developed for detecting IoT attacks depending on solutions using traditional methods to detect specific attacks. The authors summarized in Table 3.1. are proposed an IDS mechanism to detect certain attacks in IoT. Therefore, in summarized Table 3.1. the authors (Patel & Jinwala, 2021), (Bediya & Kumar, 2020) and (Kasinathan et al., 2013), proposed an IDS mechanism to detect DoS attacks, as well as (Patel & Jinwala, 2021), (Kale & Bhosale, 2021), (Sahay Rashmi et al. 2018),(David et al., 2017), and (Ahmed & Young-Bae, 2016) proposed an IDS mechanism to detect BHA in the IoT. Also, (Farea & Küçük, 2021), (Nasution et al., 2020), (Moradi et al., 2019), (Labraoui et al., 2018) and (Mendoza & Kleinschmidt, 2015) proposed IDS mechanisms to detection on-off attacks. All authors in Table 3.1. are used different methods and features to detect specific attacks. This may be effective to detect a special or certain attack, but implementing an IDS mechanism for each attack, especially with advances in technology and increasing attack types, may be inefficient. It may consume devices resources as well as prevent it from being extended to detect new attacks. (Bediya & Kumar, 2020), proposed a host-based IDS mechanism. They built a host-based IDS framework that analyzes sensor power consumption to identify DoS attacks both under normal and malicious conditions in an IoT network. They performed an experiment and showed how to set up a 6LoWPAN IoT network environment of sensors in a Cooja simulator. Also, (Patel & Jinwala, 2021), proposed the 6MID-IDS mechanism. They are a proposed blockchain architecture utilizing a 6MID Micro chain, and they use features commensurate with the proposed framework to detect DoS attacks on 6LoWPAN devices with limited resources. The readable index in the Table 3.1. continues to read in the same fashion. Therefore, the specific IDS to detect certain attack is not able to assess the resource automatically and detect the anomaly for other attacks due to the limited features input that is only dependent on one attack features. Therefore, Implementation, examination, and analysis of attacks mainly help to extract general and comprehensive criteria (features/ characteristics) for evaluating all attacks, as well as help developers,

develop an IDS based on criteria capable of detecting more than one attack that does not depend on the criteria for one attack features. Therefore, implementing and analyzing the attacks is the best way and the first point to find evaluation criteria before proposing an IDS.

Table 3.1. Proposed IDS via authors to detect certain attacks

Authors/ Year	Attacks Type	Mechanism	Description	6LoWPAN
(Patel & Jinwala, 2021)	DoS	6MID-IDS	They are proposed blockchain architecture utilized 6MID Microchain.	Detect DoS attacks on 6LoWPAN devices with limited resources.
(Bediya & Kumar, 2020)	DOS	Host-based IDS	They built a host-based IDS framework which analyzes sensor power consumption to identify attacks both under normal and malicious conditions in an IoT network.	They performed an experiment and show how to set up a 6LoWPAN IoT network environment of sensors in a Cooja simulator.
(Kasinathan et al., 2013)	DoS	EU FP7-IDS	They developed a novel IDS architecture that integrates with the EU FP7 project into the network framework.	The task is to determine DoS attacks with 6LoWPAN-based.
(Patel & Jinwala, 2021)	BHA	SIEWE-IDS	They analyzed two methods, 1- SIEWE (Strainer based Intrusion Detection of Blackhole in 6LoWPAN for IoT) and 2- T-SIEWE (Trust and Strainer based ID of Blackhole in 6LoWPAN for the IoT). SIEWE and T-SIEWE statistically-based minimize the number of nodes that must be observed, by detecting and eliminating the dangerous nodes from the network.	SIEWE and T-SIEWE are Hybrid Approach to detect Blackhole in 6LoWPAN for IoT
(Kale & Bhosale, 2021)	BHA	IDS	They are proposed methods, to calculate the performance of the network, if the performance is less than the threshold	they are used AODV protocol, and MATLAB

Table 3.1. (Cont.) Proposed IDS via authors to detect certain attacks.

(Sahay Rashmi et al., 2018)	BHA	IDS	To identify topology isolation of nodes produced forward by blackhole attacks, a technique based on the idea of exponential smoothing is proposed. The next arrival time of packages at the sink node from any node in the LLN is predicted using exponential smoothing.	6LoWPAN in IoT
(Airehrour et al., 2017)	BHA	Trust-based IDS	They have proposed a new reliable routing protocol that provides comprehensive security solutions to detect blackhole attacks.	They are proposed a trust-based 6LoWPAN-RPL routing protocol
(Ahmed & Young-Bae, 2016)	BHA	Mitigation technique-IDS	They have suggested A local decision and a global verification approach. Each node begins by listening in on packets sent by its neighbors and attempting to identify anomalous nodes by their behavior. Furthermore, if a node identifies unusual nodes, it can determine if a black hole is present in the abnormal node.	6LoWPAN-RPL networks
(Farea & Küçük, 2021)	OOA	(ANNMLP-IDS)	They are proposed the approaches based on the Artificial Neural Network Multilayer Perceptron (ANN MLP)	6LoWPAN protocol stack
(Nasution et al., 2020)	OOA	Trustworthiness Management-IDS	They are proposed a Detection Algorithm via an authentication process	ESP8266 and MQTT broker
(Moradi et al., 2019)	OOA	Trust Blockchain mechanism-IDS	They are designed Blockchain for IoT networks, to mitigate On-Off attack	They are using a remote software update process as an instance to represent OOA
(Labraoui et al., 2018)	OOA	Trust and reputation systems-IDS	They have proposed O^2 Trust is using the history(misbehavior) of each node is as a reliable factor that should affect the calculation of the trust value.	New trust IDS for WSNs

3.2. Machine Learning Methods (Anomaly-Based)

Currently, many researchers try to employ ML methods to solve security issues, this technique becomes an ingenious solution. Recently researchers have turned to use machine learning methods to detect IoT attacks and they are still continuing in this area to cover the gaps. Table 3.2. summarized the previous studies to detect attacks using machine learning techniques. The authors summarized in Table 3.2 the utilization of the ML methods to detect attacks. The different datasets have different features and different results. A dataset may be used by other contributors such as DS2OS dataset (URL2), NSL-KDD dataset (URL3), UNSW-NB15 (URL4), and Bot-IoT-2018 (URL5) to employ in ML models. However, the main challenge to the examination of intrusion detection technologies and ML models that are designed or employed for the IoT ecosystem is the shortage of real datasets and extracting the features that are affected via attacks in the IoT ecosystem. The use of datasets and extracting the features related to IoT features reflects the real IoT. Most of the authors in Table 3.2 indicated that a large gap in the literature prevents the development of effective anomaly-based intrusion detection methods since labeled datasets are not readily available as well as the criteria for evaluating the attacks and extracting them. The reason behind this gap is the lack of implementation, examination, and studies, and this is one of the problems that researchers faced, as well as difficulty in implementing the attacks in the IoT to produce an updated and up-to-date dataset containing various attacks that have the characteristics of the IoT due to technological advancements. As well as the large companies that have IoT datasets that are not shared or available with research communities. The main reason for this is privacy concerns.

Both (Bedi et al., 2021) and (Hasan et al., 2019) employ machine learning. They detected attacks in IoT sensor networks using machine learning algorithms and they used Artificial Neural Network (ANN), Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT) techniques. They used the DS2OS dataset. The dataset includes the following features as criteria to evaluate attacks and detect them: Source ID, Source location, Destination Service Address, Source type, Source Address, Accessed Node Address, Accessed node type, Value, Normality, Timestamp, Destination service type, Operation, and Destination location. The types of attacks detected based on

these criteria were Denial of Service, Data Probing, Malicious Control, Malicious Operation, Scan, Spying, and Wrong Setup. Furthermore, (Al-Akhras et al., 2020) and (Alrashdi et al., 2019) are using machine learning to create a classification model for IoT networks to detect attacks using RF, KNN, and Nave Bayes Techniques. They used the UNSW-NB15 dataset. The dataset includes the following features as criteria to evaluate attacks and detect them: network traffic, which consists of 49 features. The types of attacks detected based on these criteria were DoS, DDoS, Reconnaissance, and Theft. The readable index is in Table 3.2. continues to read in the same fashion. In many studies, there is no common factor between the features of attack. It differs from one attack to another. But there is a common factor between the techniques of the solution being used the same techniques. It is very important to update the features that are affected by attacks. which can carry out all the attacks, and these features are available on all devices in the IoT. Choosing criteria (parameters/features) for evaluating attacks is very important. For example, criteria may be chosen that apply to some attacks but not to all attacks, or may not be available on some devices. Hence, the features of evaluation in the summarized studies are very different from one dataset to another.

So, according to the previous reviews in Table 3.2., various attack datasets are employed in ML-based activities, However, using the dataset may lead to its employment in more than one research and use of the same ML techniques, which leads to similar results as well as being unrelated to the characteristics and features of the IoT or the datasets that do not reflect the IoT features. Furthermore, the datasets do not include the features of 6LowPAN devices. Also, some datasets do not include network traffic. Moreover, all the datasets do not include power consumption, which is an important criterion for evaluating the attacks. Thus, the datasets may not depend on the criteria (features/Characteristics) that are applied to all attacks in the IoT ecosystem. It is crucial to select the criteria (parameters/features) for evaluating attackers. For instance, it may be decided to use criteria that only apply to particular attacks and not all attackers, or that are not compatible with all devices. So, the features of evaluation in the summarized Table 3.2 are very different from one dataset to another. Common features of heterogeneous IoT devices were selected to evaluate attacks that make them appear homogeneous and compatible with all IoT devices. In this thesis, the criteria for evaluating attacks are radio messages and power consumption, which are considered as inputs for ML and IDS alike. Therefore,

any device in the IoT has features that are radio messages and power consumption. Additionally, all the studies in the literature review utilized the ML models but without determining the lightweight and optimal models despite their importance due to restrictions in IoT ecosystems (memory, processor, and batteries) as well as the deployment of models based on criteria. Hence, working on the constrained devices features of the IoT may help in developing a custom lightweight algorithm that matches the IEEE 802.15.4 standard. Or maybe it can be added as a layer in the stack called the "detection layer". Therefore, employing machine learning models may be easy, but extracting the obvious features as input for models and determining lightweight models may be difficult.

Table 3.2. Proposed ML techniques by authors to detect attacks in IoT

Authors/ Year	Employ Ml and Papers Names	Techniques	Features	Dataset Used	Types of Attacks
(Bedi et al., 2021)	YES "Detection of attacks in IoT sensors networks using machine learning algorithm"	ANN, LR, RF, SVM, DT	Source ID, Source location, Destination Service Address, Source type, Source Address, Accessed Node Address, Accessed node type, Value, Normality, Timestamp, Destination service type, Operation and Destination location	DS2OS	Denial of Service, Data Probing, Malicious Control, Malicious Operation, Scan, Spying, and Wrong Setup
(Churcher et al., 2021)	YES "An Experimental Analysis of Attack Classification Using Machine Learning in Iot Networks"	KNN, SVM, DT, NB, RF, ANN, LR	Stime, Sport, Dport, Pkts, Bytes, Ltime, Seq, Dur, Mean, Sum, Min, Max, Spkts, Dpkts, Sbytes, Dbytes, Rate, Srate, and Drate	Bot-IoT-UNSW	DoS HTTP, Keylogging Scan, Service Scan, DDoS UDP, DDoS TCP, DoS UDP, DDoS HTTP, Data exfiltration, and DoS TCP
(Bagui et al., 2021)	YES " Machine Learning Based Intrusion Detection for IoT Botnet"	LR, SVM, RF	H stats, MI stats, HH_jit stats, HpHp stats, weight, mean, standard deviation, radius, magnitude, cov, and pcc	IoT_botnet	Mirai and Bashlite attacks

Table 3.2. (Cont.) Proposed ML techniques by authors to detect attacks in IoT

(Al-Akhras et al., 2020)	YES "Using Machine Learning To Build A Classification Model For Iot Networks To Detect Attack Signatures"	RF, KNN, Naïve Bayes	Network traffic which consists of 49 features	UNSW-NB15	DoS,DDoS,Reconnaissance,and Theft
(Thamaraiselvi & Anitha Selva Mary, 2020)	YES "Attack And Anomaly Detection In Iot Networks Using Machine Learning"	RF, NB, SVM, DT	Ts, uid, id_orig.h, id_orig.p, Id_resp.h, id_resp.p, proto, service, duration, orig_bytes, resp_bytes, conn_state, local_orig, local_resp, missed_bytes, history, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes, unnel_parents	IoT-23 dataset	Mirai,Torii,Trojan Gagfyt,kenjiro,okiro,hakai,mushtik,Hide and seek
(Rani & Kaushal, 2020)	YES "Supervised Machine Learning Based Network Intrusion Detection System for Internet of Things"	RF	buffer overflow, perl, rootkit, loadmodule, Satan, ipsweep, portsweep, nmap, multihop, guess password, ftp write, imap, phf, warezmaster, spy, and warezclient	NSL-KDD and KDDCUP99	DoS, Probe, R2L, U2R
(Hasan et al., 2019)	YES " Attack And Anomaly Detection In Iot Sensors In Iot Sites Using Machine Learning Approaches "	ANN, LR, RF, SVM, DT	Source ID, Source location, Destination Service Address, Source type, Source Address, Accessed Node Address, Accessed node type, Value, Normality, Timestamp, Destination service type, Operation and Destination location.	DS2OS	Denial of Service, Data Probing, Malicious Control, Malicious Operation, Scan, Spying, Wrong Setup

Table 3.2. (Cont.) Proposed ML techniques by authors to detect attacks in IoT

(Alrashdi et al., 2019)	YES "AD-IoT: Anomaly Detection of IoT Cyberattacks In Smart City Using Machine Leaming"	RF	Network traffic which consists of 49 features	UNSW-NB 15	DoS, Reconnaissance, Shellcode, Generic, Analysis, Backdoor, Exploits, Fuzzers, and Worms
(Diro & Chilamkurti, 2018)	YES "Distributed Attack Detection Scheme Using Deep Learning Approach For Internet Of Things"	ANN	buffer overflow, perl, rootkit, loadmodule, Satan, ipsweep, portsweep, nmap, multihop, guess password, ftp writes, imap, phf, warezmaster, spy, and warezclient	NSL-KDD and KDDCUP99	DoS, Probe, R2L, U2R

3.3. Thesis Approaches and Techniques

In many ways, this thesis differs from previous works on IoT security in the following ways: implementation, simulation, attack types, dataset generation, parameterization (features), used protocols, and ML techniques. Table 3.3. illustrates the gaps from previous studies that are covered by the thesis.

Table 3.3. The gaps from previous studies are covered by the thesis.

Gaps / **Differences Factors**	**Gaps in Previous Studies (Anomaly-Based)**	**The Thesis Covered the Gaps in Previous Studies**
ML Techniques	They employed ML techniques.	We employed ML techniques via AMLS.
Attacks Implementation	They did not implement any attacks.	We implemented Three attacks via Contiki OS.
Attacks Analysis	They did not analyze any attacks to explore the criteria affected via attacks.	We analyzed attacks to explore the criteria affected via attacks.
Dataset Contribution	They used A dataset provided via other contributors	We contributed a dataset called KoÜ-6LoWPAN-IoT dataset.

Table 3.3. (Cont.) The gaps from previous studies are covered by the thesis.

Datasets Employment	Most datasets employment in more than one research and use of the same ML techniques, which leads to similar results.	The KoÜ-6LoWPAN-IoT dataset is the first version that was implemented in our study and research with unique results.
IoT Features-Based	Most datasets are synthetic data that do not reflect the IoT features and not updated to date.	Our dataset reflects the IoT features that are based Cooja simulator.
6loWPAN Features	The datasets do not include the features of 6LoWPAN devices.	Our dataset includes the features of 6LoWPAN devices.
Network Traffic	Some datasets do not include network traffic.	Our dataset includes network traffic.
Power Consumption	All the datasets do not include power consumption features.	Our dataset includes power consumption features.
Features Selection	Features were chosen that apply to some attacks but not to all attacks.	Network traffic and power consumption features are available in all IoT devices.
Attacks, Protocols, and ML Types	The types of attacks, protocols, and machine learning techniques differ from one study to another.	Our work is different from all previous studies in terms of the types of attacks, protocols, and machine learning techniques.
Lightweight Model	All studies did not determine the lightweight model.	We explored the lightweight models.
Deployment Model	All studies did not determine the deployment model in IoT networks.	We have demonstrated the deployment model in IoT networks.

Therefore, this thesis consists of two main phases: Thesis problems simulation and analysis phase and Thesis problems solution techniques phase. In simulation and analysis phase, we selected and implemented three different attacks on the IoT via Cooja based on the degree of difficulty in implementation, detection, popularity, spread, and severity. The selected attacks are arranged as follows: DoS, BHA, and OOA attacks. Then All scenarios are configured on 6LoWPAN protocol stack for the IoT, which consist of UDP, IPv6, ICMP6, RPL, 6LoWPAN, and IEEE 802.15.4. The reasons for choose the 6LoWPAN devices achieve interoperability, which implies that apps do not need to know the limits of the physical connections that transport their packets. It can communicate with any wireless 802.15.4 device over any other IP network connection (e.g., Ethernet or Wi-Fi).

Then we analyzed the attacks to know the parameters/features that are used as an evaluation for analyzing attacks in the scenes are radio messages and power parameters, as well as understanding the distinguished parameters of normal and malicious activities between scenarios. Furthermore, characteristics or features affected by the attacks are an obvious criterion and input for Intrusion Detection systems (IDS) or features of the datasets that can be employed in Machine Learning (ML) models. The analysis approaches for each attack consist of three-stage processes: the simulator stage, information aggregation, and extraction of the results. Then the dataset was generated based on the general characteristics that affected by any attack (power consumption and radio messages). The real dataset of IoT-based features contains a mix of data from malicious nodes and normal nodes in the IoT ecosystem. It is called the KoÜ-6LoWPAN-IoT dataset.

Table 3.4. Thesis problems simulation and analysis phase

Simulations And Implementations	**Attacks Selection**	**Attacks Types**	**Used Protocols**	**Attacks Analysis**	**Features Selection**	**Dataset Generation**
Contiki/Cooja	difficulty in implementation, detection, popularity, spread, and severity. Traffic-based, Topology-based, Trust-based	DoS, BHA, and OOA attacks.	6LoWPAN stack. (UDP, IPv6, ICMP, RPL,6LoWPAN and IEEE 802.15.4)	Analysis approaches consist of three-stage processes: the simulator stage, information aggregation, and extraction of the results.	Radio messages (84 Features) and power consumption (12 Features)	KoÜ-6LoWPAN-IoT dataset (96 Features)

In the thesis problem solution techniques phase, we employed machine learning-based approaches, which can classify more than one attack. We utilized AMLS to implement the models.

Table 3.5. Thesis problems solution techniques phase

Mechanism	Type of Algorithms	Algorithm Selection	Lightweight Model	Optimal Model	Lightweight and Optimal Model	Proposed Deployment
Machine Learning-Based Approaches AMLS	Random Forest, Decision Tree, Boosted Decision Tree, K Nearest Neighbors, SVM, ANN	Decision Tree-Based Approaches (Random Forest, Decision Tree, Boosted Decision Tree)	Decision Tree, Boosted Decision Tree, Random Forest	Random Forest	Random Forest	lightweight ML→ edge computing Other ML models→ fog or cloud computing

We proposed the decision tree-based models because the decision tree-based approach requires less computational complexity and its accuracy varies between moderate and high, which indicates to the designer and developer to design a custom lightweight algorithm for the IoT network and take the computational complexity time in mind, especially when we talked about constrained devices. Finally, we identified the optimal algorithm to detect attacks in the IoT ecosystem. Additionally, we proposed a decision tree-based model deployment and other models in IoT networks.

4. METHODOLOGY AND EXPERIMENTAL DESIGN

4.1. Proposed Model Design

This thesis depends entirely on simulation to detect attacks in IoT in all its stages to obtain the results starting from the simulation stage to reaching the stage of results. In this thesis, we divided our method into phases.

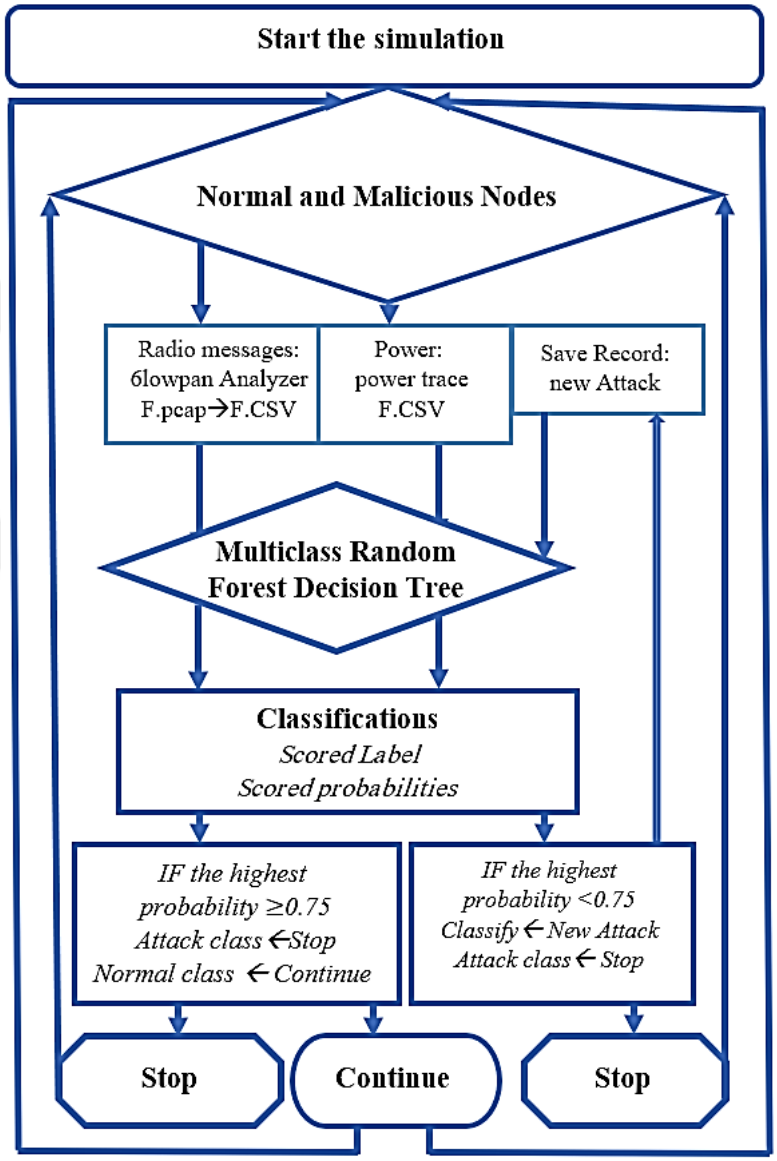


Figure 4.1. The proposed model for detecting the IoT attacks

Therefore, it consists of the simulation phase, attacks analysis phase, dataset collection and manipulation, preprocessing phase, the machine learning phase and time complexity, proposed ML-Models deployment, and the results phase. Figure 4.1 illustrates the

proposed model for detecting IoT attacks, and Figure 4.2 shows the implementation phases for the proposed model to detect IoT attacks. It starts from the simulation phase until getting results in the evaluation. Figure 4.3 depicts the procedures of the ML-based approaches to detect and prevent IoT attacks.

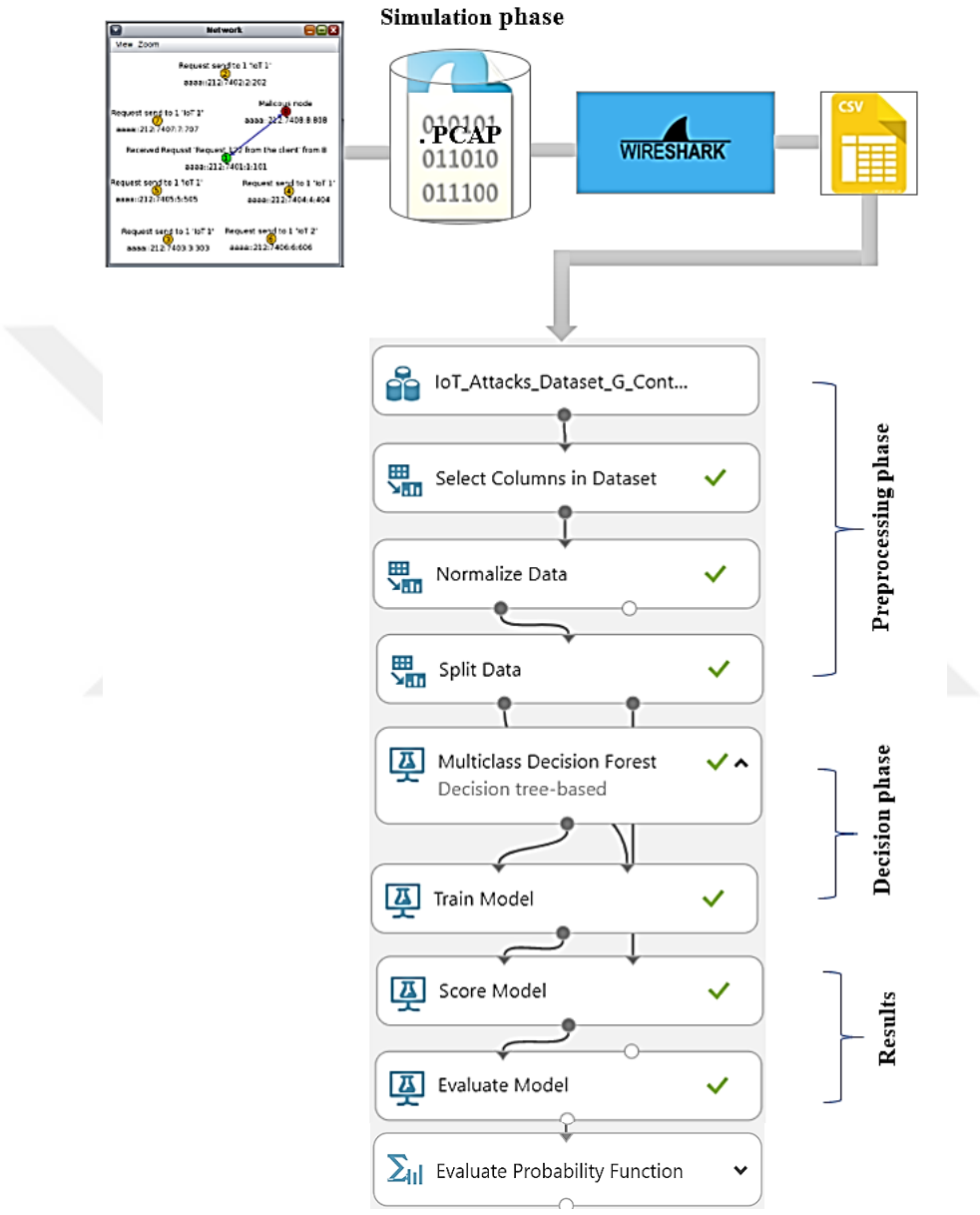


Figure 4.2. Proposed model implementation to detect the IoT attacks

Procedures: ML-based approaches to detect IoT attacks
Input: KoÜ-6LoWPAN-IoT dataset generated via Cooja simulator.
Internal processing: ML-based approaches.
Output: Performance matrix
Test Output: Probability of classes and classes label

Figure 4.3. Procedures the ML-based approaches to detect IoT attacks

```
1.  Read Features of IoT_devices (radio messages, power consumption);
2.  Store_To_ KoÜ-6LoWPAN-IoT_dataset (radio messages, power);
3.  Import_ Features (radio messages, Power);
4.  Split Data (Train (80 %), Test (20%));
5.  ML-based approaches (Multiclass Random Forest Train model (Train (80 %)));
6.  Scored model (Multiclass Random Forest (Train (80 %)), Test (20%))
7.  Evaluate_results (performance matrix, Accuracy)
8.  Test_Display_results ←( probably of classes, scored label);
9.  If (highest probably of classes>0.75 && scored label=="Attacks (DoS|| BHA|| OOA)") {
10.     Attacks occur ← Stop process of node (Close connections ());
11. End if
12. Else if (highest probably of classes>0.75&& scored label=="Normal")) {
13.     NO_Attacks ← Continue process of node;
14. End if
15. Else if (highest probably of classes < 0.75) {
16.     Save the new sample as input in KoÜ-6LoWPAN-IoT_dataset with new label name
    "New attacks"
17.     Attacks occur ← Stop process of node (Close connections ());
18. End if
19. KoÜ-6LoWPAN-IoT_datase_ check {
20. If (count of label for "Normal" > count of label for "attacks")
21.     Delete one old record are labeled Normal;
22. End if
23. Else {
24.     Delete one old record are labeled Attack;
25. }
26. End process;
27. }
```

Figure 4.3. (Cont.) Procedures the ML-based approaches to detect IoT attacks

In the simulation phase, all scenarios were implemented. This phase will be considered in detail in Section 4.2. In the attack analyzing phase, the criteria and features are explored and considered as inputs to the ML-based approaches. This phase will be considered in detail in Chapter 5. In the dataset collection and manipulation phase, the data generated during the simulation and analysis phases was collected. This phase will be considered in detail in Section 4.3, while the preprocessing phase of data will be considered in detail in Section 4.4. In the machine learning phase and time complexity, at this stage, the machine learning models that were implemented were identified and filtered based on time complexity. This phase will be discussed in detail in Section 4.5, and the filtered models' results will be extracted in Chapter 7.

4.2. Simulation Phase

We utilized Contiki OS which that includes the Cooja framework to implement the attacks. Therefore, Table 4.1. shows the general configuration in the Cooja network area.

Table 4.1. Network configuration in Cooja

Simulator	Cooja Simulator
Network area distance	190 m^2
Type of node	Sky mote
Number of nodes	7 senders and 1 root
Transport layer	UDP
Network layer	IPv6 -ICMP-RPL
Adaptation layer	6LoWPAN
Data and Physical layer	IEEE 802.15.4

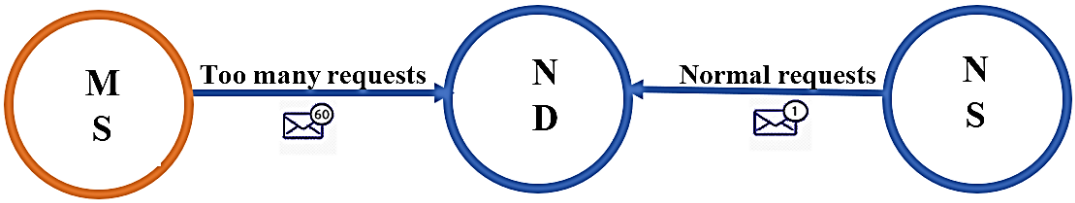
In this phase, we implemented three attacks in the IoT, based on the degree of difficulty in implementation and detection. These attacks are arranged as follows: DoS, BHA, and OOA attacks. In this thesis, all scenarios are configured on 6LoWPAN stacks for the IoT which contains protocols UDP, IPv6, ICMP, RPL,6LoWPAN, and IEEE 802.15.4. Each scenario consists of eight nodes. Node 1 is the root (server/sink). Nodes 2–7 are normal nodes, while node 8 is malicious. The color for a malicious node is red. Normal and malicious nodes either request service from the root (server) or the root (sink), which collects their data. The attacker node might be implemented on various layers. Thus, the malicious code is implemented in the transport layer as DoS, while the malicious code in BHA is implemented in the network layer and some functions are controlled in the MAC layer like duty cycle to implement the OOA in addition to the transport layer. While the simulation is running in each scenario, the radio messages and the power parameters are captured. The reason is as the malicious node directly affects the power consumption, as well as the features of radio messages, change from one attack to another during the test of attacks. As we trace the behaviors of attacks, the malicious nodes experience the most power consumption compared to normal nodes. In addition to that, the radio messages include all the matrix of 6LoWPAN protocols that we knew parameters/ features, which affect the features and parameters directly via any malicious activities. Therefore, the 6LoWPAN protocol stacks (radio messages) and the power properties are considered as criteria and inputs (dataset) to our models. The features affected by attack will be considered in detail in experimental analysis section.

The CIA triad acronym for "Confidentiality", "Integrity", and "Availability". These three terms consider the heart of security in any device as well as considered key objectives in device security. Violation of any of these three terms may put the security of the devices at a critical point and expose them to risks. Confidentiality ensures certain private or secret data is not shared or made accessible to unauthorized parties. Integrity assures a device executes its primary purpose without manipulation, free of intentional or unintentional unauthorized tampering. Also, the Availability ensures that devices work efficiently and that authorized users are not denied to access the service. DoS is one of the most well-known network-attacking behaviors. One goal of a DoS attack is to make the server unavailable to use from neighboring nodes. This means that DoS attacks violated one objective of computer security called availability. This makes device security at a critical point. If the server or resources are unavailable or destroyed to use, therefore, the malicious and normal nodes will drop their packet. The black hole attack is considered a form of denial-of-service attack when we talk about the CIA triad where the malicious node drops all the packets from neighboring nodes and make normal nodes unable to reach the intended destination. therefore, the black hole attacks violated one objective of computer security called availability as in denial of serves attacks. The ON-OFF attack breaks the integrity of devices due to inconsistent behavior and manipulation of data. Therefore, the ON-OFF attacks violated one objective of computer security called integrity. It is necessary to enhance these key objectives in computer security to make the devices more confidentiality, integrity and available via a proposed approach to address the attacks that exploit them. So, more detail will be considered about the attacks that violated the CIA triad or key objectives in computer security in Sections 4.2.1,4.2.2 and 4.2.3 respectively. There is no problem with confidential data where the 6 LoWPAN provides the encrypt the data. Therefore, we focused on the attacks that affect the availability of resources as in the DoS and BH attacks as well as the integrity of data as in the ON-OFF attack.

4.2.1. DoS Attacks

Denial-of-service attacks are a critical point in IoT devices due to constrained devices. So, DoS attacks are designed to make a machine or network resource unavailable to their users (clients, senders, etc.). In IoT, A DoS occurs when the attacker sends too many

requests to the main server/host, making the real users of the server unable to use it. The attacker node is used to flood either traffic or requests, which causes the network traffic to overflow, preventing normal requests and traffic from entering the network. Also, the malicious node indirectly prevents other nodes from gaining access to the server. The malicious node will try to deny any normal node access to the node that is attacked. Figure 4.4. illustrates the basic structure and concept of DoS attacks.



M: Malicious node, **N**: Normal node, **S**: Source, **D**: Destination

Figure 4.4. Basic structure and concept of DoS attacks

The node in red is the attacking node, which sends many requests amounting to 60 requests every minute to the node in the middle, which is the victim node, while the node on the right side is the node that sends normal requests amounting to one request every minute.

To implement the DoS scenario, in the network area, we set up eight nodes distributed circularly as shown in Figure 4.5. Node 1 is a server, and nodes 2–7 are normal nodes, while node 8 is the malicious node colored in red. All the normal and malicious nodes (clients) send their requests to the server, and the server responds to them as in Figure 4.5. As mentioned in the first remark, the blue line between the server (node 1) and the malicious node (node 8) represents the radio traffic. The flooding of radio traffic between the server and the malicious node is obvious due to the many requests from the malicious node. In other words, the number of UDP packages produced by the malicious node is extremely high when compared to the normal nodes.

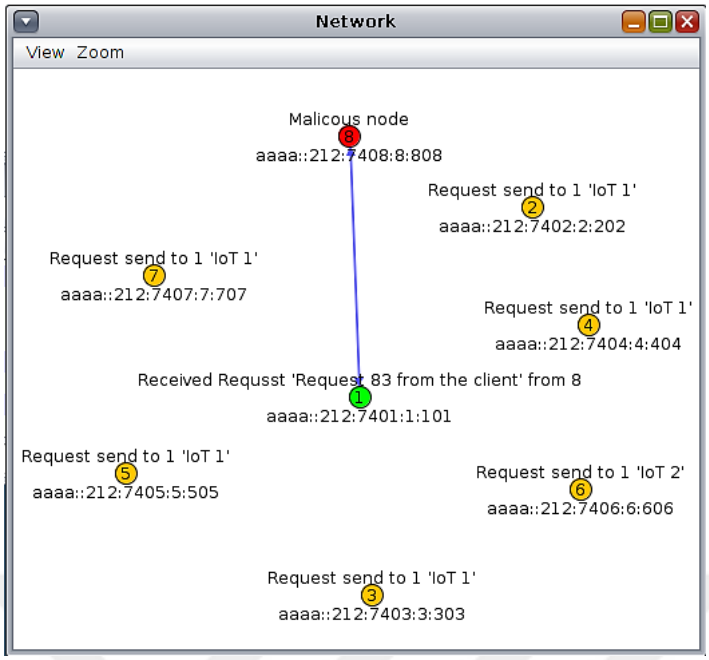


Figure 4.5. DoS scenario

In this scenario, two periods of time are defined to send requests to the server, one for the normal nodes and another for malicious nodes. The normal nodes send requests (UDP packets) to the server every minute (normal case), while the malicious nodes send their requests to the server every second. In normal nodes, the timer sends the requests within a predefined 1-minute period to their target (server), while in the malicious node, the timer is set to send the requests within a predefined 1 second period. The code of DoS attack implemented in ~/contiki-2.7/examples/IoT-attacks/DoS-attacks/udp-client.c. The structure of the DoS attack code will be considered in detail in the appendix section. Figure 4.6. illustrates the node's output and the requests made between clients and servers. In the output node window, it consists of three tabs: time, ID, and message. The time represents the node time events and the ID is referred to as the Node ID, whereas the message carries the request with the destination ID and the number of requests that have been requested. Where we notice that malicious node 8 sends too many requests to the server (node 1), the server may respond to it. We could observe in the second line that the DoS attacks are active and the malicious node sends a request to the server at 44 minutes and 48 seconds (44:48) and the server may receive it.

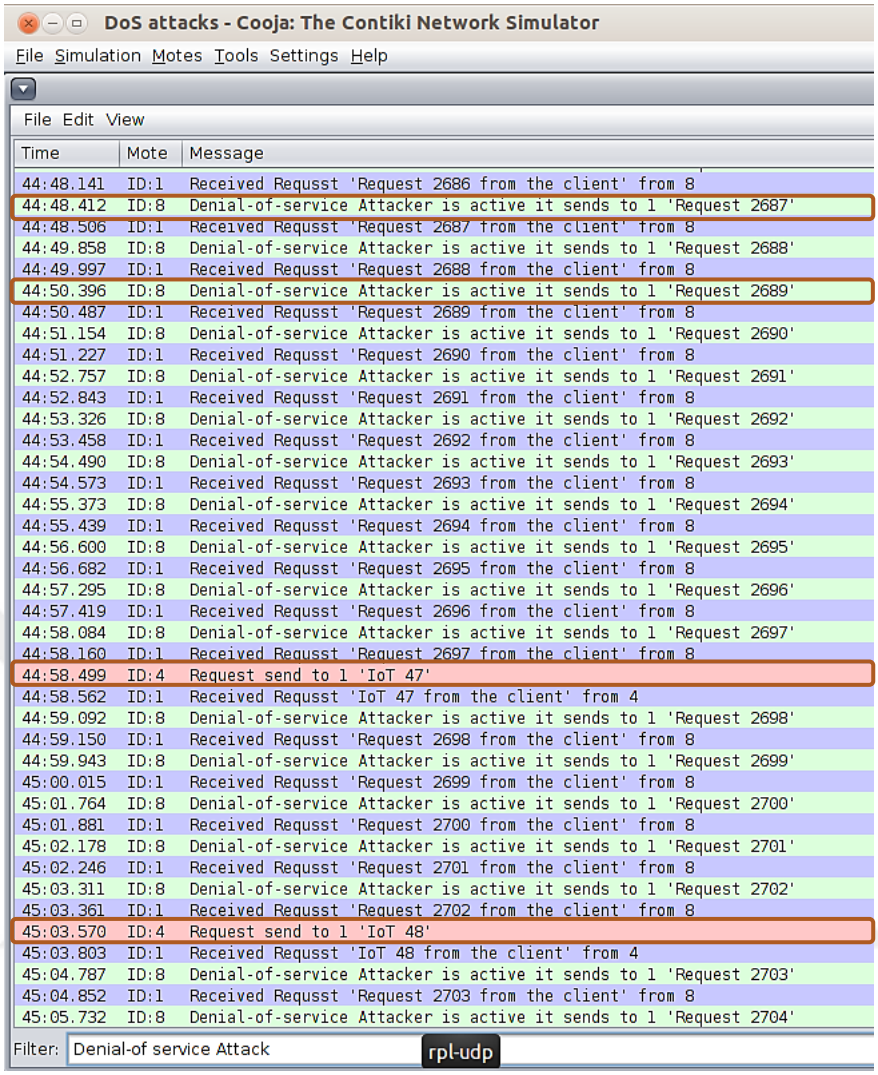


Figure 4.6. DoS nodes output

Also, the malicious node sends a request to the server at the time of 44:49. The number of malicious node requests reached 2687 requests at the time 44:48, compared to normal node 4 it reached 47 requests at the time 44:58. This means the requests from malicious nodes are sent every 1 second and the process is continuous to send flooding of requests compared to the normal nodes. For example, node 4 sends a request to the server at 44:58 and sends it again at 45:03, which means normal nodes send their requests to the server every 1 minute, and this is normal case if there is no delay in the buffer. The features affected DoS attack will be considered in detail in experimental analysis Section of the Dos attack.

As we mentioned above, the power consumption is affected directly by the malicious activity, and this leads to consuming the power. The power may be consumed if the

wireless channel is open, that is, if it is in the transmitting (radio transmit) or listening (radio listen) case, or it may be consumed by any other factor like the CPU. In Dos attacks, the malicious node consumes a lot of energy while it is in the transmission state (radio transmit) as in Figure 4.7., colored red, and the reason is that the transmission rate in the malicious node is excessively large compared to normal. For example, the malicious node in DoS attacks transmits 60 packets over one minute, while the normal node may transmit only one packet or may not if packets are delayed in the queue. In the malicious node, we note that too little power is consumed while the radio is listening and also from the CPU. They are colored yellow and green, respectively, as in Figure 4.7. It consumes too much power if the radio is in a transmitting case due to many transmit via it.

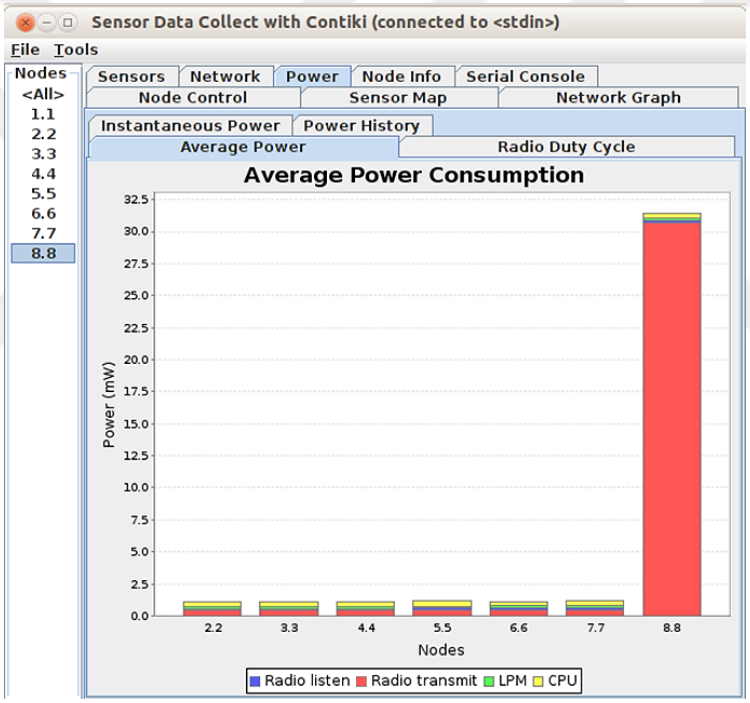
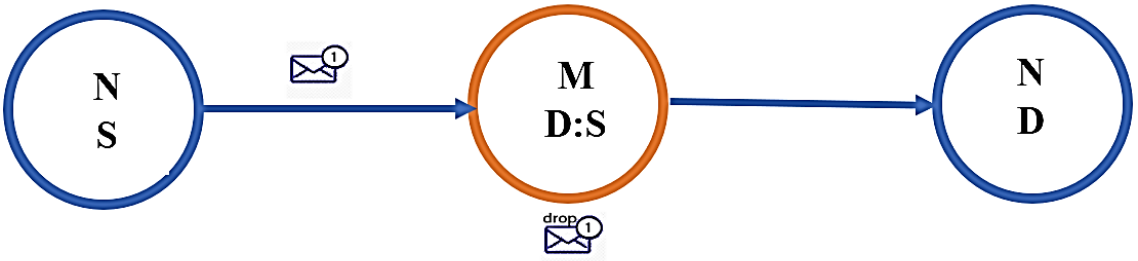


Figure 4.7. Power consumption in a DoS attack

4.2.2. Blackhole Attacks

In a special case of black hole attacks, the malicious node drops some data packets while others are forwarded successfully. This is called selective forwarding. In another case, the malicious node does not forward any data packets and discards all packages. This is called a "complete black hole attack." When implementing the special case of black hole attacks, the topology of the network remained un-isolated because the malicious node is still forwarding some packets to other nodes, whereas the malicious node in complete black

hole attacks must be isolated explicitly as in our scenario. In this scenario, we have implemented a complete black hole attack that isolates several nodes on the topology. Figure 4.8. illustrates the basic structure and concept of BHA attacks. The node in red is the malicious node, which drops or discards any packages that are received from the left-side node and does not forward these packets to the target in the right-side node.



M: Malicious node, **N**: Normal node, **S**: Source, **D**: Destination

Figure 4.8. Basic structure and concept of BHA attacks

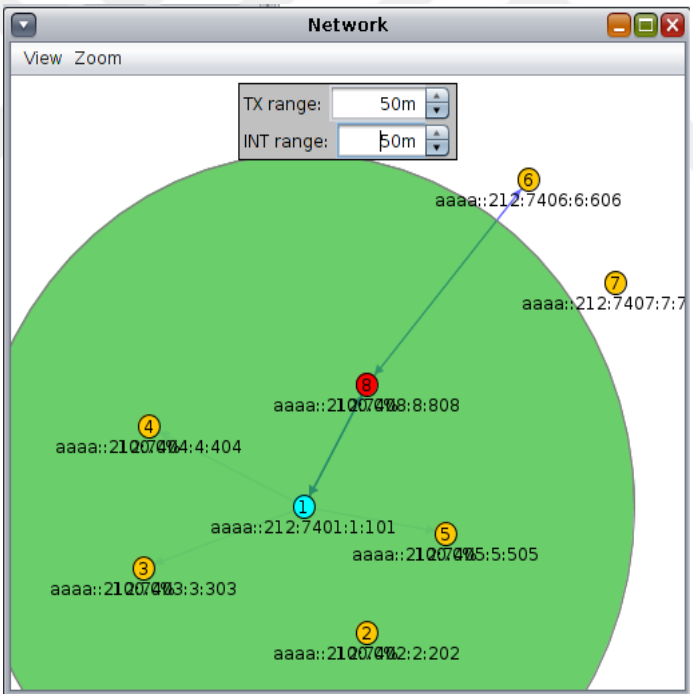


Figure 4.9. BHA scenario

The main target in BHA scenario is the sink node collects data from senders whereas the senders send their packets to the sink node. In the scenario of a black hole, we set up eight nodes. Node 1 is a sink node, where node 8 is a malicious node and the other nodes are normal. We used a multi-hop node because the black hole is more effective and affects the topology of the network. In this scenario, we placed the malicious node in a strategic

position that separates several nodes that communicate with the sink via the malicious node. For further information, some nodes are located in the direct range of the sink node, while others are not, and data packets from nodes outside of radio coverage are routed through other nodes to the sink node. All data packets from senders' nodes are destined to the sink. As appears in Figure 4.9., Nodes 2, 3, 4, and 5 are located within radio coverage, while nodes 6 and 7 are outside the radio area. The malicious node is the link between the nodes inside the radio area and the nodes outside the radio area of the sink node. The route of data packets from nodes 6 and 7 to the sink is passed via a malicious node. And this leads the topology of the network to isolate several nodes due to the malicious node in a strategic position as appear in Figure 4.10. Each one executes and implements the malicious code in its way to examine and test its intended purpose. As well, we can implement the malicious code in different layers. In this case, we implemented and developed the malicious code in the network layer.

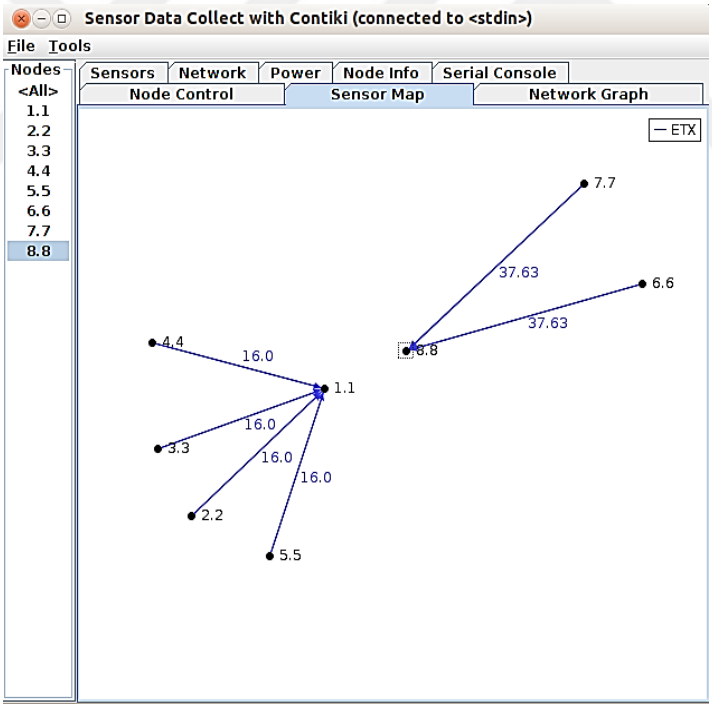


Figure 4.4. Effect BHA on topology

In the malicious node of the blackhole scenario, we set some global variables (parameters) to zero to drop all packets like uip_len, uip_flags, uip_ext_len, and uip_ext_bitmap. The uip_len variable is the length of the packet in the uip_buf buffer, and the uip_buf buffer puts incoming packets in it, whereas the uip_flags variable is used for communication

between the IPv6 and the application program like UDP. Therefore, many operational routers may be set to discard all packets with a hop-by-hop option header (HBH), but major difficulties still exist (RFC8200). In IPv6, we can have extension headers. If present, the HBH must be processed before forwarding the packet (Contiki team). In this scenario of black holes are present. As well as extension headers, HBH can be processed or handled by each node along a packet's delivery path until it arrives at its destination. These parameters are set up and configured in the malicious node to drop all the packets by the created drop function. The code of BHA implemented in ~/contiki-2.7/core/net/uip6.c. The structure of the BHA attack code will be considered in detail in the appendix section.

Thus, what's going on is that there are some conditions on the global variable. For example, if the global variable is greater than or equal to zero, it goes to the drop function. In the drop function, we return these variables to equal zero. Because these variables must usually be greater than zero due to the updating and changing of their values during each process. This means when these values of variables are updated or changed to any values, the malicious code returns these values to zero. Hence, the malicious node may or may not continue to receive and forward the data packets generated by other nodes. Also, the malicious node may not continue to process its generated packet. The malicious node drops the incoming and outgoing packets. The effects of the malicious node on the topology of the network are completely isolated. The global parameters in IPv6 were more effective in isolating the route between the malicious node and the destination. because these variables are usually change. The features affected BHA attack code will be considered in detail in experimental analysis Section of the BHA attack.

In black hole attacks, the malicious node, as well as several nodes, consumes a lot of power. For example, nodes 6 and 7 are very power-consuming, although they are normal nodes. But they are affected by a malicious node (Node 8), that drops packets without forwarding them to their destination (Node 1). Generally, the malicious node and nodes 6 and 7 consume a lot of power in both cases when the radio signal listens or transmits. We noted that nodes 6 and 7 consumed so much power due to the many packages that may be sent to form the topology and stability of the network. The malicious node consumes a lot of power, and the rate of consumption power in listening radio signals is

higher than the rate of consumption power in transmission radio signals because the dropped packages do not forward or transmit to their destination.

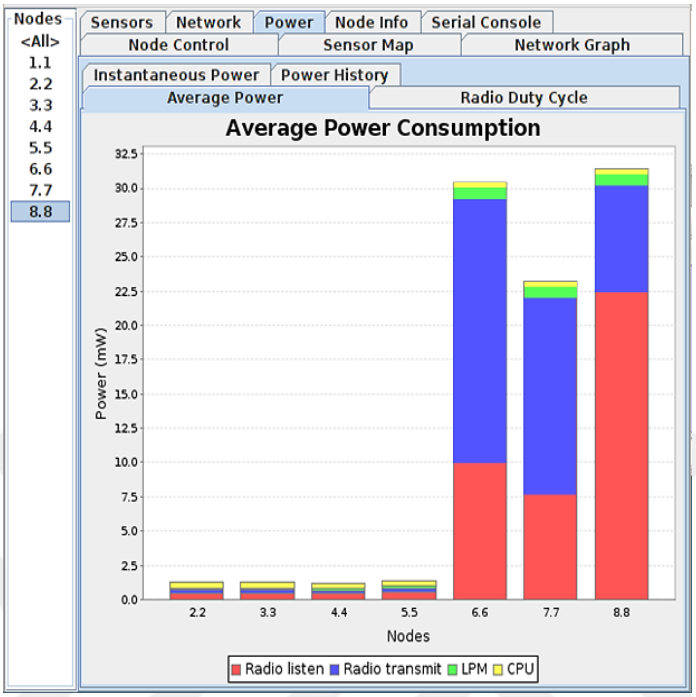
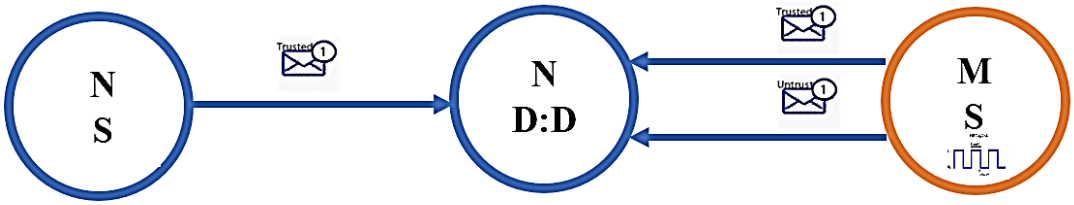


Figure 4.5. Power consumption in a BHA attack

4.2.3. ON-OFF Attacks

OOA is a sort of selective attack to avoid it being classified as an untrusted node (Mendoza & Kleinschmidt, 2015). So, the malicious node switches its behavior from harmful to normal and back again, allowing it to remain unnoticed while launching attacks. Therefore, in this attack, there are two statuses: the ON status is called "attack," It is a critical case while the OFF status is called "normal.". This attack hits the advantage of the dynamic features of trust by exploiting the time-domain of status and inconsistent behavior (Farea & Küçük, 2021). The attacker swaps between ON and OFF. When the cycle is ON, the malicious node initiates attacks, and when it is OFF, it does nothing (Nasution et al., 2020) and (Farea & Küçük, 2021). An OOA attacker often has to deal with various neighbors to gain incompatible opinions of trust from the same node. Figure 4.12. illustrates the basic structure and concept of OOA attacks. The node in red is the malicious node, which alternately generates trusted and untrusted packets to their destination, which is the middle node.



M: Malicious node, **N**: Normal node, **S**: Source, **D**: Destination

Figure 4.6. Basic structure and concept of OOA attacks

The middle node receives suspicious and unreliable packets due to receiving inconsistent packets from the malicious node. See the output of the malicious node (node 8) in Figure 4.14 which alternately produce trusted and untrusted packages and sends them to neighboring nodes.

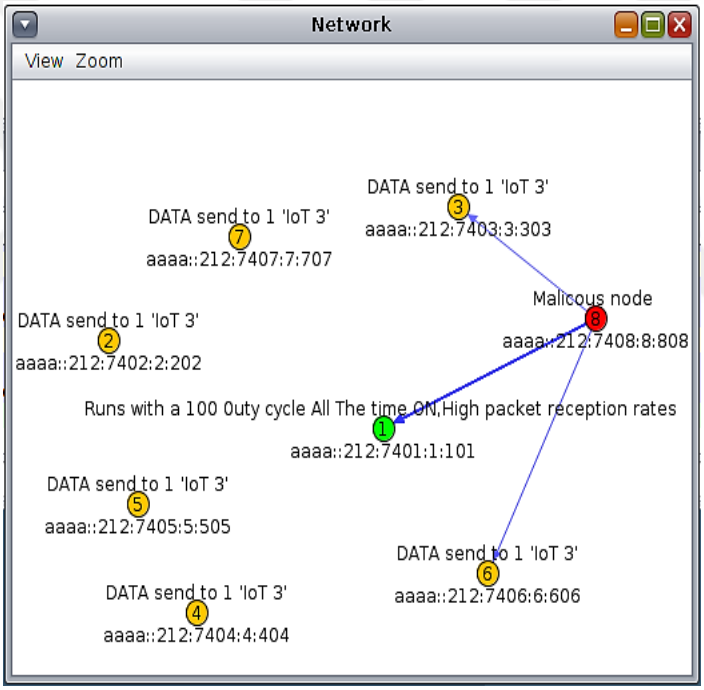


Figure 4.7. OOA Scenario

ContikiMAC is a radio duty cycling technique that employs periodic wake-ups to monitor neighboring packet streams. If a packet transmission is detected during a wake-up, the receiver is kept turned ON (Dunkels et al., 2011) so that the packet may be received. When a packet is received correctly, the receiver sends an acknowledgement. The transceiver (transmitter-receiver) must be completely turned between OFF and ON to send and receive radio if the status is ON and save power if the status is off. Therefore, to achieve low power consumption, ContikiMAC nodes sleep most of the time and

intermittently wake up to check for radio activity (Dunkels et al., 2011). The purpose of this scenario is to achieve the properties of an OOA to create a real dataset that will be trained in our models to detect OOA. The malicious node in our scenario switches its action from the attacker to normal and from normal to attackers if the case malicious node's cycle is ON via sending both trusted and untrusted packages randomly.

In this scenario, we initialized eight nodes that were placed randomly in the network area as shown in Figure 4.13. Node 1 is a server. Node 8 is malicious, while the other nodes are normal. Node 1 is exposed to any malicious node because this node is always active (ON-status). This node wants to receive its data from neighbors' nodes in a position of trust, without any doubt. Node 8 is a malicious node that generates and sends inconsistent data alternatively. In a malicious node, the duty cycle of the ON status to OFF status is set up similarly, fifty-fifty percent. The reason is that it makes it easier to detect malicious behavior if the radio status is ON. Normal and malicious nodes request a service from the server. The malicious code was implemented only in node 8. This malicious node alternately produced trusts and untrusted packages to node 1. Node 1 was set up to have an ON status all the time. It is not known if the data being sent by the malicious node is reliable or unreliable data.

In a malicious node, we created two functions. The first function is called "Trusted_Function," and the second function is called "Untrusted_Function." The Trusted_Function sends trusted packages to node 1, while the Untrusted_Function sends untrusted packages. In another sense, the Untrusted_Function sends UDP packets that are not standard and have different parameters (i.e., payload) than their peers. Additionally, this function sends their packages at an abnormal time. For example, all the normal nodes send their UDP packets every 1 minute. The trusted function in malicious nodes also sends their UDP packets every 1 minute. Unlike the Untrusted_Function, it sends its UDP packets every 30 seconds. Thus, the malicious node did these functions, generating trusted and untrusted packages randomly to put neighboring nodes in doubt with their inconsistent behavior. Node 1 is the service provider for the other nodes. This node is vulnerable to any malicious node because it is always in the ON status (active). It is more vulnerable to attacks from malicious nodes. Because it receives huge packets from nodes, especially from malicious nodes. Node 1 may bring the malicious code from malicious

nodes and spread them to other nodes. The code of OOA implemented in ~/contiki-2.7/examples/IoT-attacks/ON_OFF-attacks/udp-client.c. The structure of the OOA attack code will be considered in detail in the appendix section.

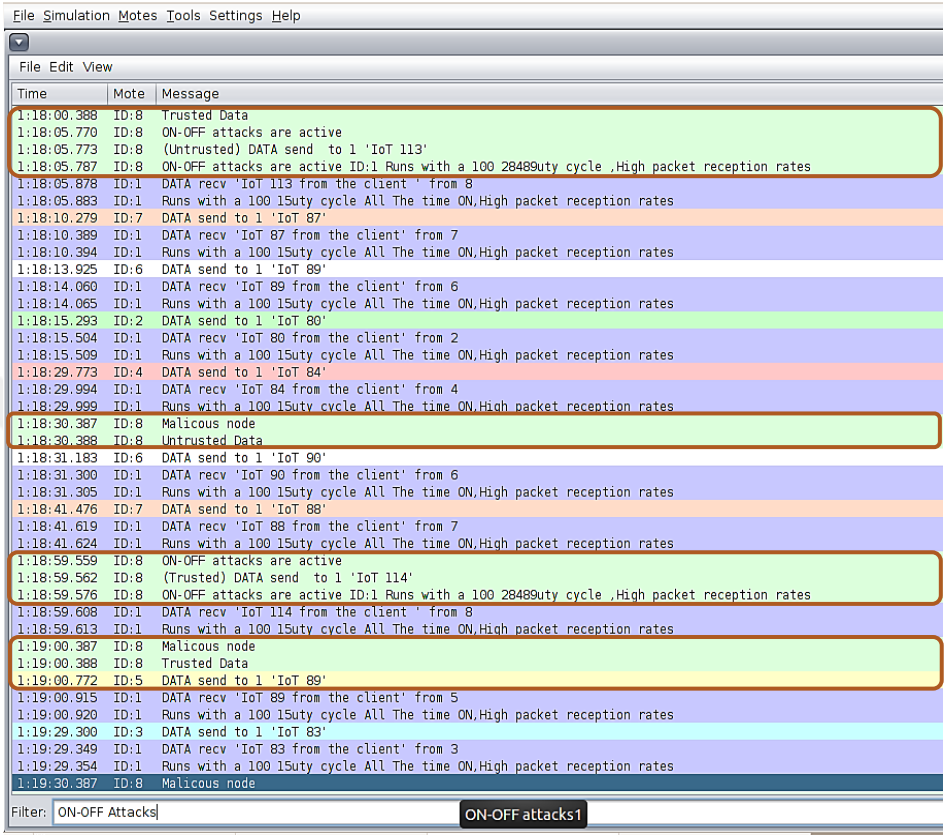


Figure 4.8. OOA mote output

We use the implementation of ContikiMAC as Radio Duty Cycling (RDC) in the MAC layer. ContikiMAC is a duty-cycling mechanism that allows nodes to keep their radios off as much as possible to achieve low power consumption and save energy (Dunkels et al., 2011). The default radio duty cycling mechanism in Contiki 2.7 (Dunkels et al., 2011) uses a power-efficient wake-up mechanism with a set of timing constraints to allow devices to keep their transceivers off. By default, this setting is active when we initialize nodes in the Cooja network area. Thus, the function of the duty cycle is that we can set it up however we want. This function was called in the OOA attacks scenario to serve the purpose of keeping the malicious node active and node 1 at risk. Alternately, the malicious node switches to lunch the malicious code (Untrusted_Function) and normal code (Trusted_Function). It remains undetected while the status is active. As for the node that is vulnerable to attack, the parameters of this function were set to 1 (100%), which

means node 1 has its radio transceivers ON, high reception packets, and power consumption. The parameters for the malicious node were set to 0.5 (50%). Figure 4.14 illustrates the output of nodes in the OOA scenario. This shot was captured after spending over an hour in the Cooja simulator. The features affected OOA attack will be considered in detail in experimental analysis Section of the OOA attack.

The power consumption in ON-OFF attacks is different from previous attacks. In ON-OFF attacks, power is consumed when the radio transmits signals or listens to them. Furthermore, the CPU consumes high power due to a change in status from ON to OF at an unusual period of time as in Figure 4.15 If the CPU consumes much power, it may lead to an increase in the temperature of the device, which is considered a dangerous case because it may damage the device.

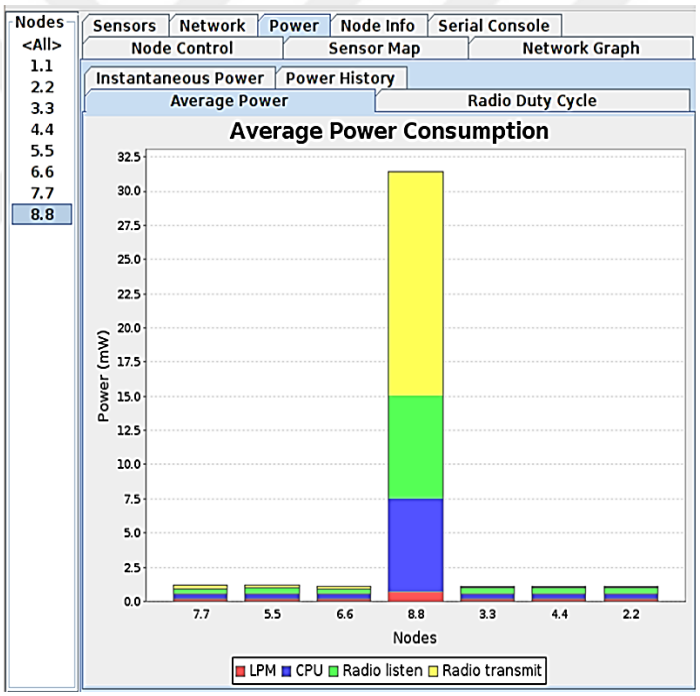


Figure 4.9. Power consumption in a OOA attack

4.3. Dataset Collection and Manipulation

At this stage, we collected all the datasets generated via Contiki OS that was captured by the 6LoWPAN analyzer and Power Trace tool. Table 4.2. summarizes the number of observations/samples and the number of features that were collected and processed that represent the radio messages and power features in each scenario. Figure 4.16. shows the

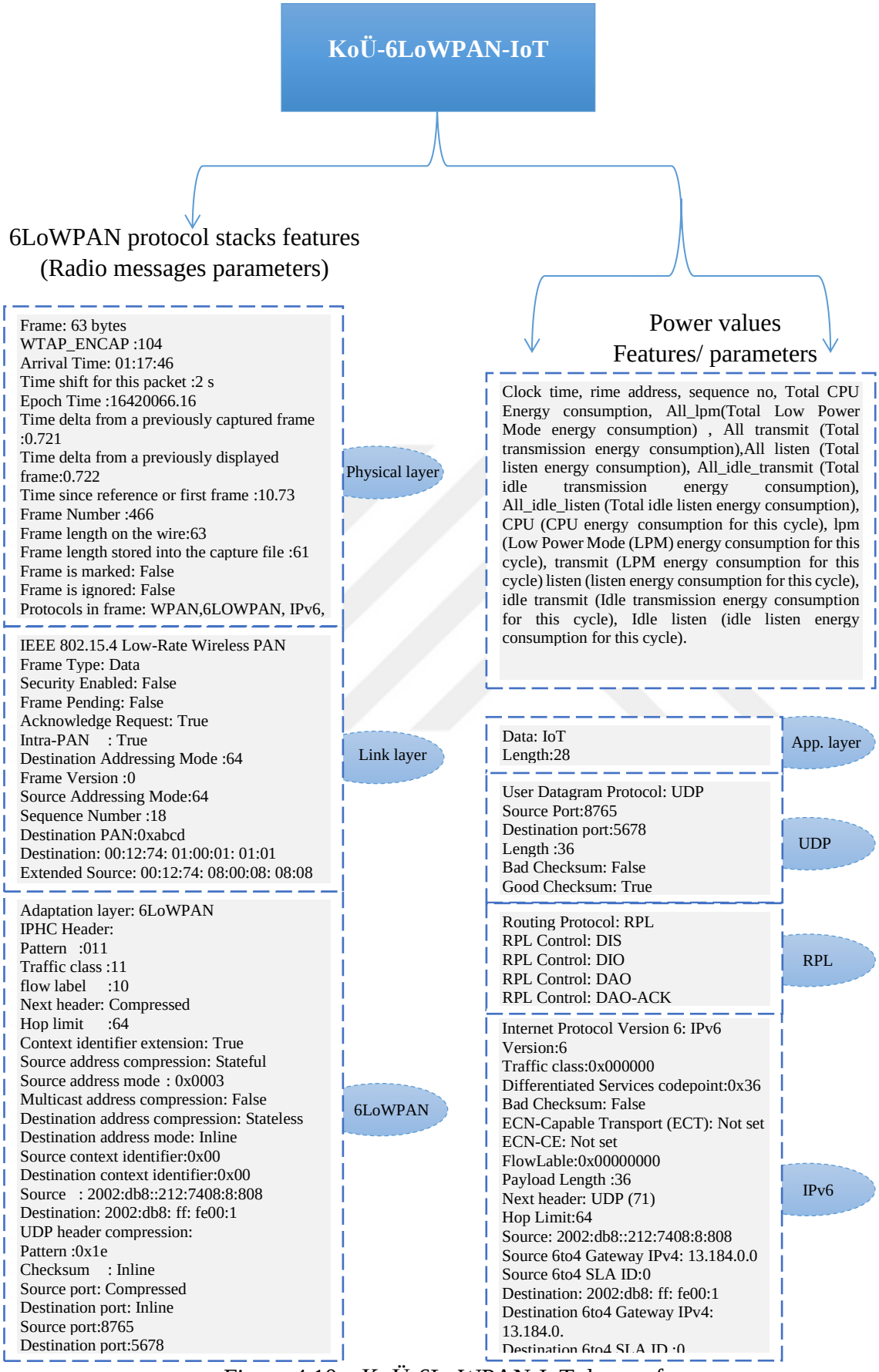


Figure 4.10. KoÜ-6LoWPAN-IoT dataset features

KoÜ-6LoWPAN-IoT dataset features extracted from Wireshark and the power app via Cooja. The analysis of the radio message improves its security, and the analysis of the power improves its reliability. From all the scenarios, we obtained three datasets. The numbers of captured data are 9912, 12696, and 25072 from DoS, BHA, and OOA, respectively. These three datasets are merged into one dataset and are called the KoÜ-6LoWPAN-IoT dataset of IoT attacks. In the process of merging the datasets at the beginning of each dataset, a sample of 20% was taken and copied one by one into the KoÜ-6LoWPAN-IoT dataset from each attack dataset, and then the remainder was copied successively into the file that represented 80%. The datasets are not directly merged one by one. The reason is that in the ML phase when we split the data into training and testing data, the testing data was taken from the DoS sample only because the dataset was large and the testing dataset was not randomly extracted from all attack samples.

Table 4.2. Observations and features in each attack

Attacks types	Simulation duration	Samples of Radio message and power trace	Number Features of Radio message/ power trace
DoS	45 min	9912	84 /12
BHA	1 hour	12696	84/12
OOA	2 hours	25072	84/12
Total	3:45	47680	96

4.4. Pre-processing Phase

The practice of preparing raw data for use in a machine learning model is known as data preprocessing. It is the first and most critical step in the development of a machine learning model. Real-world data, in most circumstances, has noise, missing values, and is in an unsuitable format that cannot be directly used for machine learning models. Therefore, data preprocessing is a necessary step in manipulating data and preparing it for a machine learning model, as well as improving the model's accuracy and efficiency. The first thing we need to develop a machine learning model is a dataset because a machine learning model is completely dependent on data. A dataset is the collection of

data for a certain topic in the appropriate format. Like the KoÜ-6LoWPAN-IoT dataset that was generated via Contiki OS to detect malicious activity and inconsistent behavior.

The train-test split procedure is a very important part of Ml. It is used to train and estimate the performance of machine learning algorithms. Thus, we split the dataset into 80% for training and 20% for testing. The IoT dataset contains a huge number of samples of attacks collected from DoS, BHA, and OOA attacks, which total 47680 samples of observations, 84 features from radio messages, and 12 from the power trace. The number of samples that were captured in DoS is 9912 samples over 45 minutes, while the number of samples that were captured in BHA is 12696 samples during 1 hour and the number of samples that were captured in OOA is 25072 samples over 2 hours. Therefore, the number of samples for training was 38142, while the number of testing was 9535. The features contain a mixture of categorical and numerical data.

4.5. Machine Learning Based Approaches and Time Complexity.

Enabling ML techniques to solve various issues in IoT is a very strong point where ML solves issues in IoT such as: routing (ex: finding the optimal dynamic path), MAC (efficient channel assignment), congestion control (predicting congestion locations), event detection (detecting an event from complex sensor data), energy consumption, and so on. Therefore, here we used it to address the security of IoT. At this stage may many algorithms operate in different ways that may lead to the same goal and results. Therefore, it can be difficult to choose the right algorithm, especially in light of devices resource constraints.

In this section, we investigated to find lightweight and optimal models for the machine learning model based on time complexity and accuracy. Therefore, an algorithm that has less computational complexity and high accuracy been supposed to be chosen for implementation. Since we identified the attacks label and executed them, so we employed supervised machine learning algorithms. We compared the time complexity and accuracy of the most popular machine learning algorithms demonstrated in Table 4.3. Therefore, to find the lightweight model, we need to calculate the time complexity for ML algorithms. In time complexity we tried to find the worst case. The worst-case scenario is the best case for evaluating the algorithms or models. The optimal model means the

model that achieves high accuracy in worst case. The higher the degree of time complexity and the closer the model is to the exponential, the more the model is heavy and needs more processing, and consumes hardware resources. The devices may not be able to process data and need a device with large processing capabilities. If the complexity time is low and close to constant, the model is light in weight, requires less processing, and does not consume hardware resources. We used a worst-case time complexity calculation through which we can evaluate machine learning algorithms and models. We neglected to calculate the time complexity of the best case through which the optimal solution can be reached quickly, and this does not give us the correct evaluation and results in all cases.

We utilized the features of the KoÜ-6LoWPAN-IoT dataset that was generated by the Contiki OS, which contains 96 properties and 47680 observations. The formulas were extracted from (URL6), (URL7), and (URL8). Time complexity determines the lightweight model, while the accuracy determines the optimal model. In this study, we seek to explore a lightweight and optimal model. Table 4.3. compares the computational complexity time and accuracy of the six top ML algorithms.

In Table 4.3. the time complexity for training models for RF, DT, and BDT algorithm are 2.7863 s, 0.055727 s, and 2.7863 s respectively. This gives us an indication that the time complexity for training of these algorithms is very small in seconds compared to others the time complexity for training models of the K Nearest Neighbors, SVM algorithms, and ANN are 9.31 min, 15.4136 h, and $211*10^7$ s respectively. Therefore, the time complexity for training models of Decision Tree Based has time complexity less than KNN, SVM, and ANN. The decision tree based includes RF, DT, and Boosted Decision Tree (BDT) which has the structure of a tree and takes its decisions based on the output of the tree. Therefore, the decision tree-based models are lightweight models than KNN, SVM, and ANN.

The time complexity for testing models of the decision tree (DT) is a very lightweight model, which is 0.096 μs, which is the shortest time between models compared to the KNN model, which is a very heavyweight model, which is 0.915 ms, which is the longest time between models. We noted that the DT occupies a space in memory up to 12 KB, which is the lowest value among the other models, followed by RF and BDT, occupying

a space of 600 KB for both models. Compared to other models, ANN occupies the highest space, up to 130.95 GB, followed by KNN and SVM, occupying a space of 446.9 MB for both models. Therefore, the decision tree-based models (DT, RF, and BDT) are more lightweight models than KNN, SVM, and ANN, and they occupy less space.

Accuracy is the achievement of the optimal model. The accuracy was extracted from many previous studies on a different dataset. Therefore, we considered the results above 90 as achieving the optimal accuracy models, while the results range from 70 to 90 represent the moderate model's achievement. Otherwise, they are low models. Therefore, most of the models' accuracy results range from high to moderate.

In this thesis, we preferred four machine learning algorithms that have a tree structure and make their decisions based on a tree (Decision Tree-Based: Multiclass Random Forest, Multiclass Decision Tree, Random Forest regression, and Boosted Decision Tree). The main reason for this is that these algorithms have practically proven their efficiency in accurately outputting results and working on the KoÜ-6LoWPAN-IoT dataset produced by Contiki OS, which is difficult for some machine learning algorithms to deal with until these categorical data are converted into numerical data using one-hot encoding or any other method. This leads to an increase in the number of features after converting them, which leads to a slow model in efficiency. Furthermore, the algorithms that do not work on decision tree-based models need more time to finish training than algorithms that work on decision tree-based models. For example, ANN and SVM need a lot of time to finish training the data on it. The AMLS takes more than an hour and 15 minutes to train the model on 47680 samples. The reason is that ANN and SVM are more computationally expensive to finish training, unlike the algorithms that work on decision tree-based systems that are less computationally expensive to finish training as in Table 4.3. Also, when a model is tested to predict and classify new data, the response time of the model that has been trained on the decision tree-based approach is much faster compared to models that do not work on it. For example, when the random decision forest model was tested, the response time of the model was most up to 4.8 µs, while the ANN model needed more than 28.8 µs in AMLS platform.

Table 4.3. The computational time complexity for the top ML algorithms

n=Number of training samples (38142), n#= Number of testing samples (9535), M= number of features (96), T= number of trees (50), D=depth of tree (96), L=layers (3), S= number of support vectors (4), K= number of neighbours (4), N=neurons (100)				
ML algorithms	**Computational Complexity**			**Accuracy**
	Train Time Complexity(ns)	**Test Time Complexity(ns)**	**Space Complexity (Kbit)**	
RF	O(T*nlog(n)*M) 50*38142*Log (38142) *96 2786335874=2.7863 s	O(M*T) 96*50 4800= 4.8 µs	O(T*D) 50*96 4800=600 KB	**98 %** (Farea & Küçük, 2022) **98%** (Bedi et al., 2021), **99%** (Thamaraiselvi & Anitha Selva Mary, 2020), **95%**(Churcher et al., 2021), **100 %** (Bagui et al., 2021), **100 %** (Al-Akhras et al., 2020)
DT	O(n*log(n)*M) 38142*log (38142) *96 55726717.48=0.05572 s	O(M) 96=0.096 µs	O(D) 96=12 KB	**0.87 %** (Farea & Küçük, 2022) **96%** (Churcher et al., 2021), **96 %** (Thamaraiselvi & Anitha Selva Mary, 2020), **99%** (Bedi et al., 2021), **98%** (Rani & Kaushal, 2020)
BDT	O (T* nlog(n)*M) 50*38142*log (38142) *96 2786335874=2.7863 s	O(M*T) 96*50 4800=4.8 µs	O(T*D) 50*96 4800=600 KB	**80 %** (Farea & Küçük, 2022)

Table 4.4. (Cont.) The computational time complexity for the top ML algorithms

KNN	$O(K*n^2*M)$ 4*381422 96 5.586*1011 = 9.31 min.	O(n#*M) 9535*96 915360=0.91 5 ms	O(n*M) 38142*96 3661632=446.9 MB	**99 %**(Churcher et al., 2021), **100 %**(Al-Akhras et al., 2020), **0.96%** (Rani & Kaushal, 2020)
SVM	$O(n^3)$ 381423 5.5489*1013=15.413 6 h	O(S*M) 4*96 384=0.384 µs	O(n*M) 38142*96 3661632=446.9 MB	**98%** (Bedi et al., 2021), **79%** (Churcher et al., 2021), **99%** (Bagui et al., 2021), **99%** (Thamaraiselvi & Anitha Selva Mary, 2020)
ANN	$O(n^4)$ 381424 $2.11*1018 =211*10^7$ s	O(L*M*N) 3*96*100 28800=28.8 µs	O(n*L*M*N) 38142*3*96*100 1098489600=130.9 5 GB	**99%**(Farea & Küçük, 2021)**,** **99%** (Bedi et al., 2021), **74%** (Churcher et al., 2021),

Table 4.3. compares the computational complexity and accuracy of the six top ML algorithms, where it is observed that the decision tree algorithm has less computational complexity and the accuracy range between high and moderate, while the ANN has more computational complexity and the accuracy range between high and moderate. The random forest has less computational complexity with high accuracy, so it might be a fair model and a lightweight algorithm to detect attacks in the IoT. In general, the decision tree-based approach requires less computational complexity, and its accuracy varies between moderate and high which is indicated to the designer and developer to design a custom lightweight algorithm for the IoT network and take the computational complexity

time in mind during designing especially when we talked about constrained devices. Generally, the decision tree-based algorithm has less computational complexity, and the model's accuracy ranges between high and moderate. Therefore, this type of model achieves lightweight models and optimal models. While other models in Table 4.3. have more computational complexity and the accuracy ranges between high and moderate. Therefore, this type of model achieves heavyweight models and optimal models. Also, Finding the lightweight and heavyweight models gives us a decision to deploy models that will detect attacks and which may be in cloud or fog or edge computing. Hence, we will consider the edge, fog, and cloud computing in details in the Section 4.6.

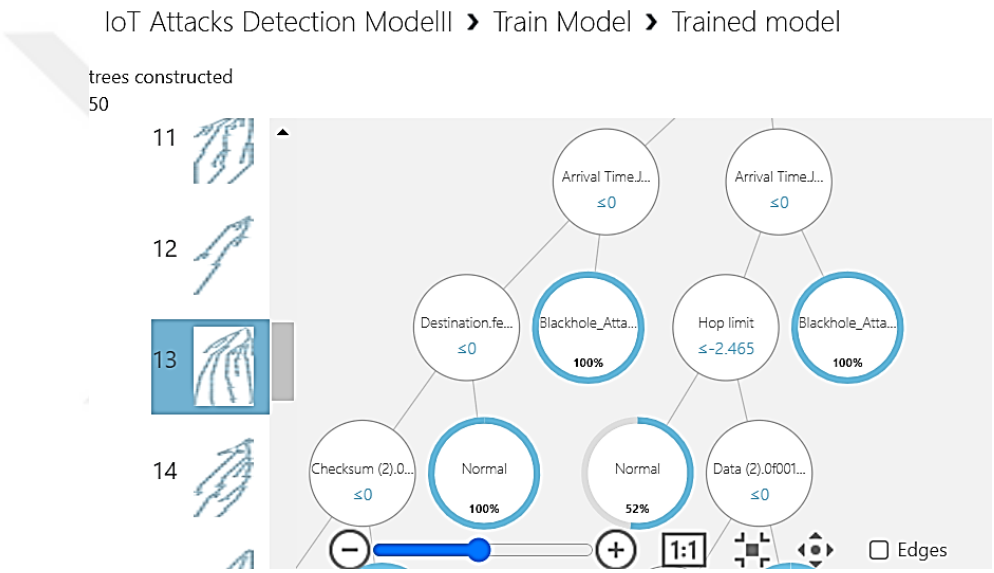


Figure 4.11. Snippet for decision tree-based Model

The decision trees, random forests, and boosted decision tree methods are similar with minor changes. A decision tree is a set of sequential stages that answer a question, make a certain decision, and provide decision probabilities, while a random forest uses the decision tree algorithm in a randomized fashion.

It consists of different decision trees that are different in sizes and shapes and combines their decisions using averages or majority rules at the end of the process. Where the boosted decision tree takes their combined decision using averages or majority rules at the beginning, instead of at the end. Figure 4.17 illustrates the structure of the decision tree-based model that is used to detect attacks in the IoT.

4.6. Proposed ML-Models Deployment

In this section, we investigated to propose the models' deployment in the IoT ecosystem network. The models used to detect the IoT attacks may be in the cloud, fog, or edge computing. Cloud computing is the most popular method of analytics and processing of IoT data. Fog and edge computing have also been widely employed to speed up and improve data processing and bring intelligence closer to IoT devices that produce data and act on it. Cloud computing can process and store massive amounts of data. Hence, if the model is deployed in cloud computing, we can employ any ML model that achieves higher accuracy to analyze and detect the attacks in the IoT regardless of the time complexity and lightweight model. Unlike edge computing, it is necessary to take into consideration the time complexity and lightweight model as well as an optimal model with the best accuracy due to device constraints. Hence, the edge computing offers attractive features where the latency for the model is very low and the privacy is very high, unlike cloud computing which is monitored by a third party. The challenge for edge computing is to find a lightweight and optimal model. Table 4.4. illustrates the features are offered via edge, fog, and cloud computing.

Table 4.5. The characteristics offered via edge, fog, and cloud computing

ML-Models Deployment	Data Analysis	Access Approaches	Latency	Internet-Based	Privacy
Edge	Process the data and make the decision in real time (at the edge)	TinyML or lightweight Model (internal processing)	Low	No	High
Fog	The fog is an intermediate layer between the cloud and the edge. It decides to process the data locally or send it to the cloud (network close to the edge).	Nodes send Encrypted data to fog computing (External processing)	Medium	Yes	Moderate
Cloud	less time-sensitive data processing (away from the edge)	Nodes send Encrypted data to cloud computing (External processing)	High	Yes	Low

Based on Table 4.3. the decision tree-based model requires less computational complexity and its accuracy varies between moderate and high. Therefore, the decision tree-based model space will be suitable for edge computing, unlike models that require more computational complexity like ANN and SVM. The model will be suitable for cloud computing or fog computing. Any ML model implemented in edge computing can be implemented in fog and cloud computing, but the opposite is not always true. Because the edge requires a lightweight model that does not consume device resources.

In this context, we recommend using the decision tree-based model to implement on the edge to detect attacks in the IoT because it is a lightweight and optimal model. The edge computing can't able to process massive data unlike cloud and fog computing can able to process big data. For cloud and fog computing, we recommend using any ML model that achieves high accuracy regardless of the time complexity. Because the cloud and fog are able to process and store massive amounts of data, it doesn't matter if resources are consumed. We tend and prefer to process the data internally on the edge as much as possible, especially in the security of the IoT, in order to avoid disclosure of data privacy by a third party.

5. ATTACKS EXPERIMENTAL ANALYSIS

5.1. Attacks Analysis Method

In this thesis, our analysis approaches for each attack consist of three-stage processes: The simulator stage, information aggregation, and extraction of the results. In the simulation stage, six scenarios have been implemented. In the aggregation stage: the tools needed to collect the data were used, such as the 6LoWPAN analyzer, Cooja.log, and power trace application to aggregate the data. Data extraction stage: It is the stage in which the characteristics and features affected by the attacks were extracted and compared with the attack-free scenarios. In the simulation phase, we utilize Contiki OS to implement the IoT attacks. Contiki OS is a powerful stimulator for the next generation IoT ecosystems. Therefore, we have implemented three normal scenarios as well as three malicious scenarios that include DoS, BHA, and OOA in IoT. Therefore, these types of scenarios were created to record and compare the parameters/features for them. In normal scenario we implemented it free of malicious nodes, and we used it as a benchmark for comparison. The malicious scenario is including the malicious nodes. Therefore, in each scenario, the normal scenarios (NS) are free from the malicious node while the malicious scenarios (MS) include a malicious node in each attack. Figure 5.1. shows the analysis approach. In the simulation phase, we have implemented six scenarios. Three of these scenarios are free from any malicious node and the other three scenarios include the malicious node. three scenarios that do not include the malicious node consider a benchmark for evaluating the other three scenarios that include the malicious nodes. All the scenarios are set up on the 6LoWPAN protocol which contains protocols UDP, IPv6, ICMP, RPL,6LoWPAN, and IEEE 802.15.4. In the malicious scenarios, each scenario has eight nodes in total. The (server/sink) is the root and the node number is 1. Nodes 2 through 7 are free from malicious nodes, while node Eight includes the malicious code. In this study, the parameters/features that are used as an evaluation for analyzing attacks in the scenes are radio messages and power parameters. The reason is that when implementing any scenario and following the behavior of the harmful node, we noted that the power consumption of the harmful node is very high compared to normal nodes as in Figures 4.7, 4.11 and, 4.15, as well as the features of radio message (6loWPAN protocol stack) in malicious activities are affected due to the change and update matrix/variables

in each process. Therefore, while the simulation is running in each scenario, the radio messages and power parameters are captured.

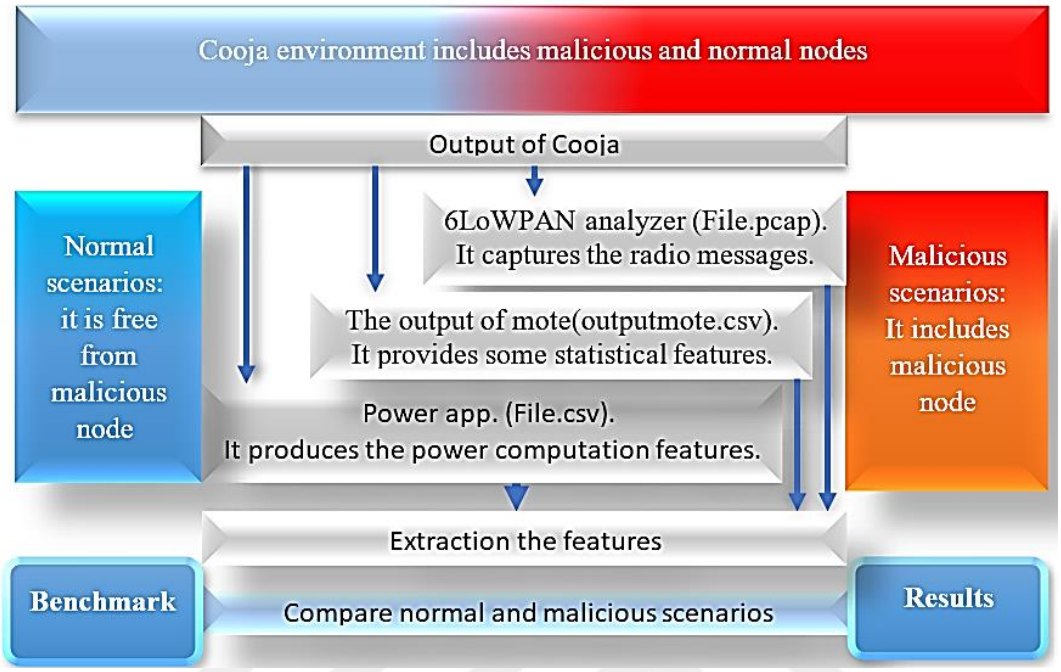


Figure 5.1. Attacks analysis method

The radio messages are captured via a 6loWPAN analyzer tool and opened utilizing Wireshark, as in Figure 5.2., where the power features are captured by calling the power trace application and displayed in the output of Cooja and saved as a file. csv for analyzing as in Figure 5.3., so this is considered data aggregation. The radio messages include all the parameters of the 6loWPAN protocol stack. while the power trace includes all related features of power per node, like transceiver power consumption, low power mode and, etc. In the extraction stage, some features that affect the network in each scenario are sorted, extracted, and then compared to obtain the results.

This Section summarizes the statistics of the observed parameters in normal scenarios and malicious scenarios, as in Table 5.1 for DoS attack, Table 5.3 for BHA, and Table 5.5 for OOA. While Table 5.2, Table 5.4, and Table 5.6 summarize the comparison between normal scenarios and malicious scenarios. Table 5.2, Table 5.4, and Table 5.6 has been extracted from Table 5.1, Table 5.3 and Table 5.5. where the normal scenario is a benchmark for the malicious scenario, the goal is to understand in depth the difference between normal and malicious scenarios and their impacts on some obvious parameters

and features. As well as Attack analysis may give us a standard approach to extract features that are affected by the attacks. These features may be used as features of the dataset that are used and tested by machine learning models, or may be used as input to the IDS to test its accuracy in detecting attacks.

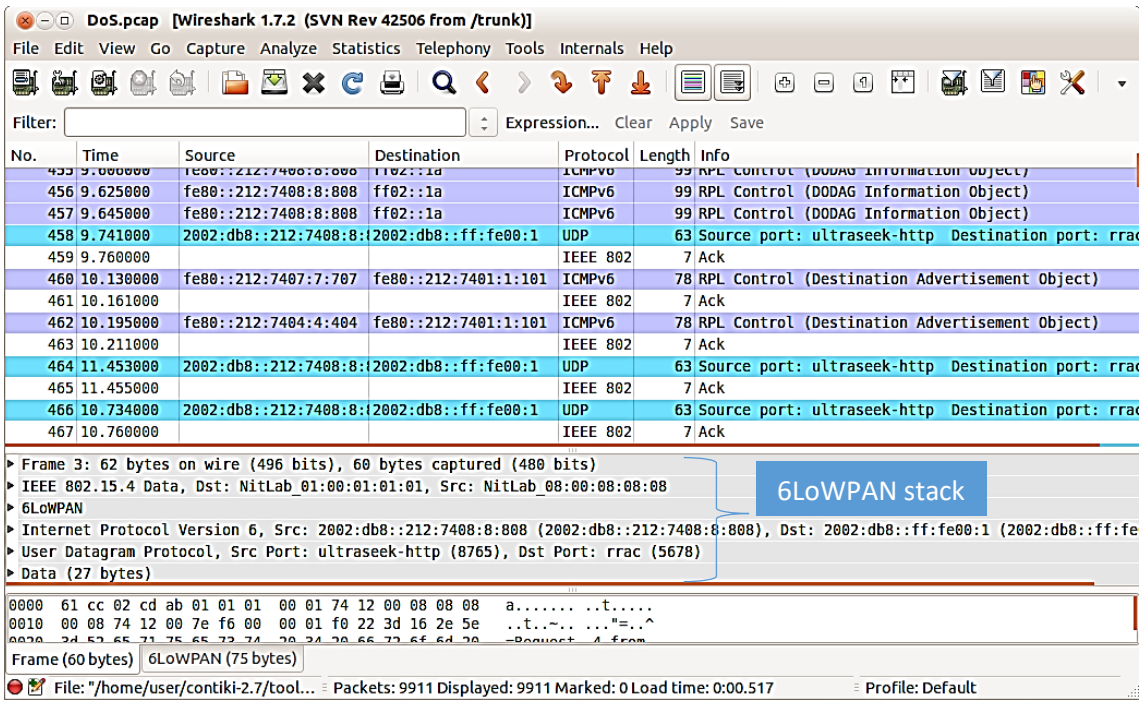


Figure 5.2. Radio messages features captured

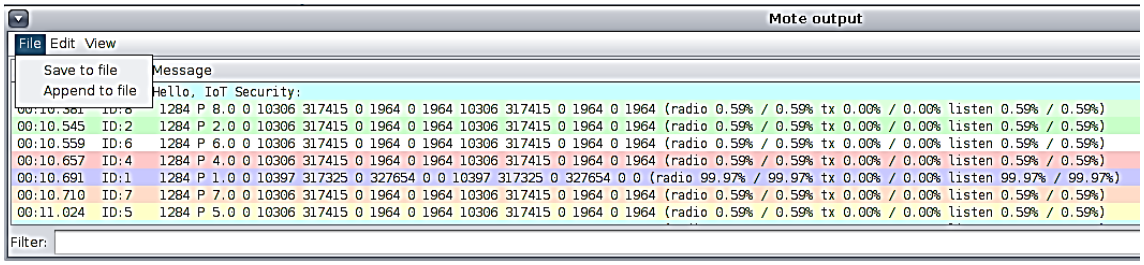


Figure 5.3. Power features captured.

5.2. DoS Attack Analysis

In these scenarios, the parameters under evaluation in the DoS situations are UDP packet sent/received and UDP packet delivery by sink node, as in Table 5.1. Packet delivery is the ratio of the number of data packets sent from all nodes to the number of data packets received successfully at the sink. The Sink node (node 1) does not release any data packets and it consumes higher power Figure 5.4. We used a counter to count the requests from

the clients to the server and traced them via output window. Another option is to use Wireshark to provide us with details about the packets.

Ideally, all normal nodes must produce 1 request every 1 minute and 45 requests over 45 minutes to the server unless there is a delay in the buffer queue. Because the normal nodes send their requests to the server every minute, so in comparison to the normal scenario and malicious scenario in DoS as shown in Table 5.1 and Table 5.2. Where the number of requests in the normal scenario was 311, while this number in malicious scenario increased by a very high rate + 866 %, reaching 3007 requests. This increase in requests from malicious/client 8 increases the load on the server eight times over compared to the normal scenario and It is capable of destroying the server or target in a few minutes. In the malicious Scenario As the overall number of requests received by the sink node from individual nodes decreased due to the number of dropped packets by malicious nodes and their delay in the buffer queue where the number of packets dropped in the malicious Scenario reached 480 This is represented as a ratio of 18 % of the total packages. The number of requests from the malicious node represents the highest numbers, reaching 2704, while the number of requests from other nodes does not exceed 60 over 45 minutes. These unexpected requests released by the malicious node to the server indirectly prevents other nodes from gaining access to the server in the expected time additionally delays occurred in receiving packages to the server node this due to the performs of the node(server) that has been attacked to work functions improperly.

For more information, the period time to send malicious node requests to the server can be decreased from a second to a millisecond, but care must be taken when executing this due to the malicious node producing huge trash data to the COOJA.log, and this lead to crash the Cooja simulation. Because the COOJA.log will be filled in a few minutes. Also, we can increase the simulation time to greater than 45 minutes but we should pay attention to the file.PCP is produced by a 6LoWPAN analyzer from time to time. Because the malicious node generates a huge data packet and the file.PCP can't open and will be damaged As in Figure 5.5. Decreasing the period time from second to millisecond makes the malicious node block the normal nodes indirectly to access the server.

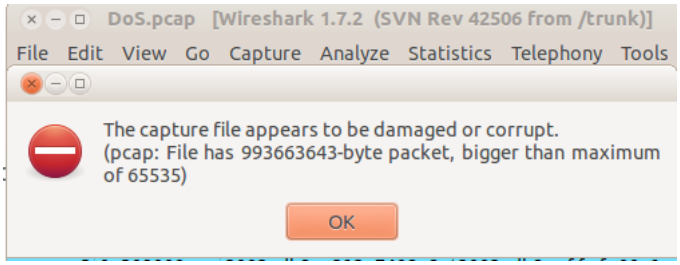


Figure 5.5. DoS PCP.File damaged

Table 5.1. Comparison between the NS and MS nodes in DoS attack

Scenarios	**Normal Scenario (NS)**		**Malicious Scenario (MS)**	
Message Type / Clients ID	UDP packets		UDP packets	
	Requests	Received	Requests	Received
ID:2	45	45	43	19
ID:3	44	43	44	44
ID:4	45	44	48	39
ID:5	45	45	58	41
ID:6	43	43	59	37
ID:7	44	43	51	43
ID:8	45	44	**2704**	1873
Total	**311**	**307**	**3007**	**2096**

Table 5.2. Benchmark for increase and decrease between NS and MS in DoS

Scenarios	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**
Message type	UDP packets Requests	UDP packets Requests	UDP packets Received	UDP packets Received	UDP Packets drop	UDP packets drop
Total	311	3007	307	2096	4 packets	480 packets
Increase /decrease	**Benchmark**	**+ 866 %**	**Benchmark**	**+ 582 %**	**0.64%**	**18 %**

5.3. BHA Attack Analysis

In RPL, various control messages are exchanged between sender nodes and sink nodes to form a topology. One important control message a DIO It is the carrier of information regarding the RPL instance and its configurations. Increased DIO messages indicate network routing topology instability, whereas decreasing DIO messages indicate a stable network. Therefore, the parameters for evaluation the BHA attack are packet delivery fraction and rate of a specific control message known as the DIO message, as illiterates

in Table 5.3, and the topology of the network, as shown in Figure 4.10. These parameters are explicit to observe. The 6loWPAN analyzer is used to capture the network traffic and inspect closely what is happening on the network. Table 5.3. shows the analysis of the network taken from Wireshark during the capture of the radio message in scenarios using a 6loWPAN analyzer.

Table 5.4 compares increase and decrease between normal and malicious scenario in BHA. Evaluating the rate and frequency of DIO messages issued by nodes, as well as the Packet Delivery Fraction in varying situations, it is clear that owing to malicious behavior brought by node 8, the rate and frequency of DIO messages published by nodes have increased. In the normal scenario, the network topology is regarded as stable, and the frequency of DIO messages reduces overall, but in the malicious scenario, the quantity of DIO messages increases overall due to all nodes experiencing an unstable network topology. In the normal scenario, the total number of DIO packages generated was 5159 but, in the malicious scenario, this number expanded by 61 % up to 8326. Also, we noted that in the normal scenario, the total packet created was 397, however, in the malicious scenario, the total packet generated jumped by 162 % up to 646. All nodes in the malicious scenarios except nodes 6 and 7 suffered a much higher delay compared to the normal scenario and the reason behind packet delay could be buffer queues. In the malicious scenario, nodes 6 and 7 generate higher UDP packets among nodes due to malicious activity introduced by node 8 because it does not return the acknowledgment to the neighboring nodes.

Table 5.3. Comparison between the NS and MS nodes in BHA attack

Scenarios	**Normal Scenario (NS)**			**Malicious Scenario (MS)**		
Message Type / Node ID	DIO messages	UDP packets		DIO messages	UDP packets	
	Released	Sent	Received	Released	Sent	Received
ID:2	710	58	58	1044	47	40
ID:3	714	59	57	1044	49	44
ID:4	805	54	54	1073	49	41
ID:5	701	60	58	986	48	40
ID:6	742	53	53	1073	219	0
ID:7	783	54	53	1073	233	0
ID:8	704	59	59	1073	1	1
Total	**5159**	**397**	**392**	**8326**	**646**	**166**

It drops all incoming packets but sends their packets until the acknowledgment is back to them. In the normal scenario, the number of packet deliveries at the sink was 392; in the malicious scenario, this number decreased by -57% and reached 166. While the sink node was creating and processing DIO packets and attempting to stabilize the network topology, packets may have been queued and waiting to be processed. In each case, the total number of packets received at the sink node is calculated. The malicious scenario had a higher number of packets dropped by the malicious node, thus a higher loss of 59% as well as this rate decrease in the normal scenario. Also, we noticed that the malicious node has only 1 UDP packet generated by it, and all nodes in a network are affected by the malicious activity as we computed above. These nodes suffered the loss despite having idealistic conditions.

Table 5.4. Benchmark for increase and decrease between NS and MS in BHA

Scenarios	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**
Message type	DIO Released	DIO Released	UDP packets Sent	UDP packets Sent	UDP packets Received	UDP packets Received	UDP packets drop	UDP packets drop
Total	5159	8326	397	646	392	166	5 packets	480 packets
Increase /decrease	**Benchmark**	**61%**	**Benchmark**	**62%**	**Benchmark**	**-57 %**	**0.63%**	**59%**

5.4. OOA Attack Analysis

In these scenarios the parameters under analysis are trusted package and untrusted package that generated by malicious and the status of radio either ON or OFF. Table 5.5 details the features for evaluation. In addition to the Table 5.6, it shows the increase and decreases values in the scenarios. In a malicious scenario, the number of trusted packages decrease by -7.6 % while the number of untrusted packages is increase reach to 100% due to the normal network are not released any untrusted packages. In a malicious scenario, the number of radio average in case the radio ON reach 1893% due to the malicious node consuming the power where the malicious node consumes 69% compared to peers at most 1.53 % as in node 7. This indicates that the malicious node either affects the features and characteristics of the tiny device that constantly updates and changes. Because the analysis of the device's features makes trustworthy more reliable and the analysis of the radio message makes it more secure.

Table 5.5. Comparison between the NS and MS nodes in OOA attack

Scenarios	**Normal Scenario (NS)**					**Malicious Scenario (MS)**				
Message Type / Node ID	UDP Sent		Radio Average %			UDP sent		Radio Average %		
	Trusted	Untrusted	ON	TX	RX	Trusted	Untrusted	ON	TX	RX
ID:1	0	0	1.22	0.11	0.94	0	0	100	1.28	10.08
ID:2	119	0	1.04	0.17	0.06	116	0	1.04	2.16	0.28
ID:3	112	0	1.04	0.17	0.07	111	0	1.04	1.13	0.40
ID:4	89	0	0.99	0.15	0.06	120	0	0.99	1.13	0.36
ID:5	117	0	0.98	0.13	0.04	113	0	0.98	2.17	0.44
ID:6	119	0	1.03	0.18	0.06	101	0	1.03	2.16	0.40
ID:7	120	0	1.53	0.55	0.03	119	0	1.53	1.02	0.29
ID:8	107	0	1.01	0.16	0.03	43	67	69.6	1.17	0.36
Total	**783**	**0**	**8.84**	**1.62**	**1.29**	**723**	**67**	**176.21**	**12.22**	**12.61**

Table 5.6. Benchmark for increase and decrease between NS and MS in DoS

Scenarios	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**	**NS**	**MS**
Message type	Trusted	Trusted	Untrusted	Untrusted	ON	ON	TX	TX	RX	RX
Total	783	723	0	67	8.84	176.21	1.62	12.22	1.29	12.61
Increase /decrease	**Benchmark**	**-7.6%**	**Benchmark**	**100%**	**Benchmark**	**1893%**	**Benchmark**	**654%**	**Benchmark**	**877.51%**

6. ANALYTICS AND SIMULATION TOOLS

6.1. Contiki OS

Contiki is a networked operating system that is designed to function on hardware with severe memory, power, and parameter constraints, with an emphasis on low-power wireless IoT devices. Contiki contains the 6LoWPAN stack network mechanisms which provide the routing protocol for low power and lossy networks IPv6 with the 6LoWPAN header compression and adaptation layer for IEEE 802.15.4 links (Contiki tutorials).

6.2. Cooja Simulator

Contiki includes the Cooja framework. Cooja is a powerful simulator utilized in the IoT. It is a network simulator designed for simulating sensor networks. It's a Java-based simulator that lets us write sensor nodes in the C language (Charles & Kalavathi, 2018).

6.3. 6LoWPAN Analyzer Tool

A 6LoWPAN analyzer is a tool built on the Cooja framework that captures radio messages and saves them as packets with an extension. PCAP was developed to capture all the packet data with details. Once this tool is activated when running the Cooja simulator and scenario, it does its job of capturing and saving the file with the extension PCAP automatically. This PCAP file can be opened with Wireshark to know the details of the packages closely.

6.4. Wireshark

Wireshark is a network packet analyzer that tries to display the packet data details (Charles & Kalavathi, 2018). It allows viewing packet data from a live network or a previously stored capture file interactively like PCAP. The PCAP format is one of the native capture file formats for Wireshark, which it can read and write. Wireshark is used to monitor network traffic and keep a close eye on what's going on in the network. It enables us to retrieve this data and convert it to CSV for further processing and analysis, as we did in our model to detect IoT attacks. By default, it displays four features of the packets, such as source, destination, protocols, and information control message. In our study, we enabled all the features and parameters for the packets, which numbered 84

features. The reason is due to the impact of attacks on some parameters in packets, which in turn make them more accurate in detecting attacks and knowing the behavior of malicious nodes than normal nodes. The features of the packet are different from one another, and the number of features is too large, thus aiding our model in detecting misbehavior in a good manner.

6.5. Power Trace Tool

Power tracing is a run-time power profiling method that estimates each node's power usage via power state tracking. By calculating the time each component spends in each power state. The ContikiMAC low power radio duty cycling mechanism is utilized by Cooja (Verma & Ranga, 2020). The goal of radio duty cycling is to switch the radio off as much as possible while still being able to communicate to save power. A node cannot receive transmissions from neighbors if the radio transceiver is turned off. To communicate while keeping the radio turned off as much as possible, the radio must periodically wake up between two statuses to receive packets from neighbors (Verma & Ranga, 2020). Nodes in a duty-cycled network do three things: transmit packets, receive packets, and periodically wake up so that it can receive packets from neighbors. These parameters that are calculated by the power traces tool are called and printed in each scenario and converted to a CSV file. Because the malicious activity in the network is affect these parameters in some types of attacks. For example, the malicious activity maybe effects on radio state and still the status ON all the time thus leading to consuming much energy and maybe vulnerability to any malicious activity. The radio must be turned completely off-duty-cycled- as possible to decrease power consumption and prevents other dangerous issues.

6.6. Azure Machine Learning Studio

Azure Machine Learning Studio (AMLS) is a platform that provides machine learning algorithms (URL9) in separate modules for creating and deploying ML workflows on Azure. It is a cloud solution that helps us speed up and manage our machine learning projects. It is a set of services and technologies aimed at assisting developers in the development and deployment of machine learning models. It may be used by machine learning experts, data scientists, and engineers in their workflows to design, train, and

manage models. Individuals and teams deploying machine learning operations inside their company may use AML to move machine learning models into production in a safe and auditable environment. We can build our experiment from scratch using one of the popular programming languages such as Python or R. In addition to that, it provides ready-made modules that make it easier for us to build and test our model, which contains practical and common modules and algorithms in artificial intelligence. In this study, we utilized an MLS because it provides a flexible and extensible framework for machine learning. Each stage of this process is handled by a different type of module, which may be updated, added to, or eliminated without impacting the rest of the experiment.

7. RESULTS

Most of the results and observations (comments, notes) or findings of this work are discussed in detail in chapters 4 and 5. This chapter will demonstrate model results that use ML-based approaches for detecting attacks in the IoT ecosystem. Especially the decision tree-based approaches that release the final decision to detect IoT attacks. The ML decision tree-based models were able to analyze and evaluate data by predicting and classifying them into normal and malicious nodes as in Figure 7.1. In classification decision tree-based, the evaluation model gives us the parameter values of estimation that are derivative values from the confusion matrix. Therefore, the overall performance of the confusion matrix for attacks is illustrated in Figure 7.2. In detail, the performance of the confusion matrix for multiclass decision random forest and multiclass decision tree models is shown in Tables 7.1 and 7.6 respectively.

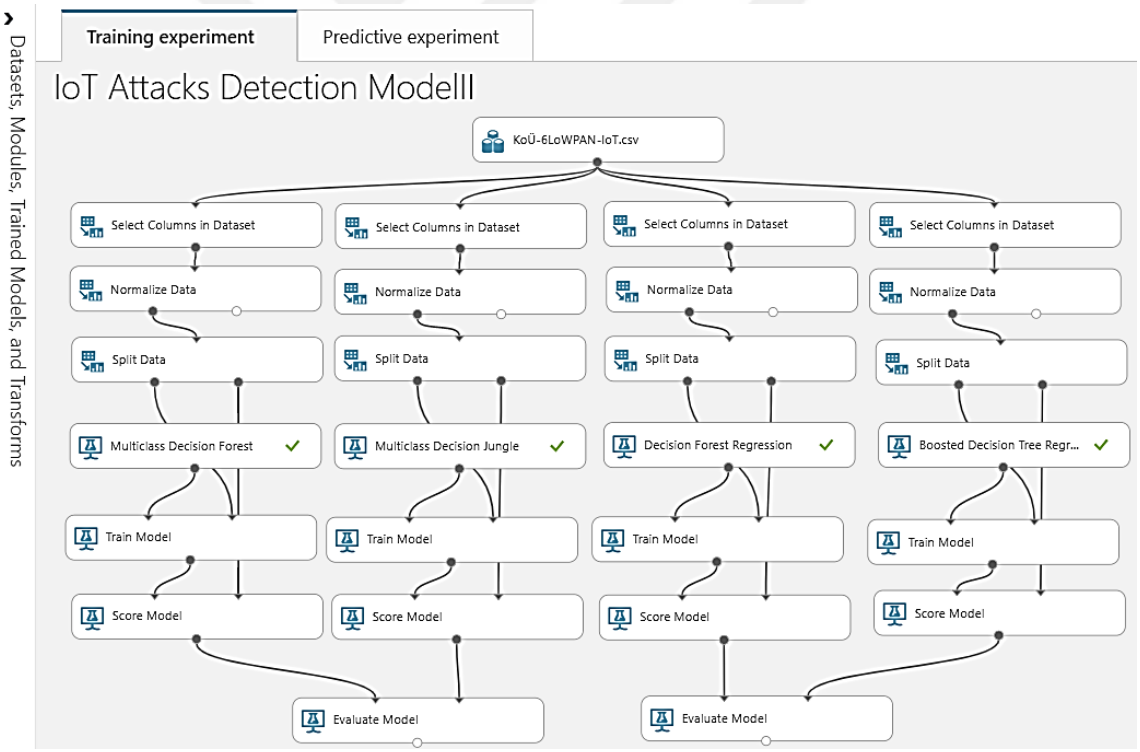


Figure 7.1. Decision tree-based models

Hence, in the multiclass random decision forest ML-based model, we obtain the overall accuracy, averaged precision, and averaged recall of 98.9%, 98%, and 97.1%, respectively. In comparison to multiclass decision tree, we get 82.5%, 82%, and 44.1% for overall accuracy, averaged precision, and averaged recall, respectively.

In regression decision tree-based, the metrics to evaluate the models are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Relative Squared Error (RSE), and Coefficient of Determination (CD). Therefore, the overall performance of the regression model's matrix for attacks is illustrated in Figure 7.3. Tables 7.11 and 7.12 detail the performance of regression models matrix for decision random forest regression and boosted decision tree regression model, respectively. Hence, in the decision forest tree regression, is carried out at 0.138 and 0.143 for MAE and RMSE while achieving 6.75 % and 93.2 % for RSE and CD, respectively. In comparison to boosted decision tree regression, we got the MAE and RMSE of 0.12 and 0.246, while for RSE and CD we obtained 19.83% and 80.1%, respectively.

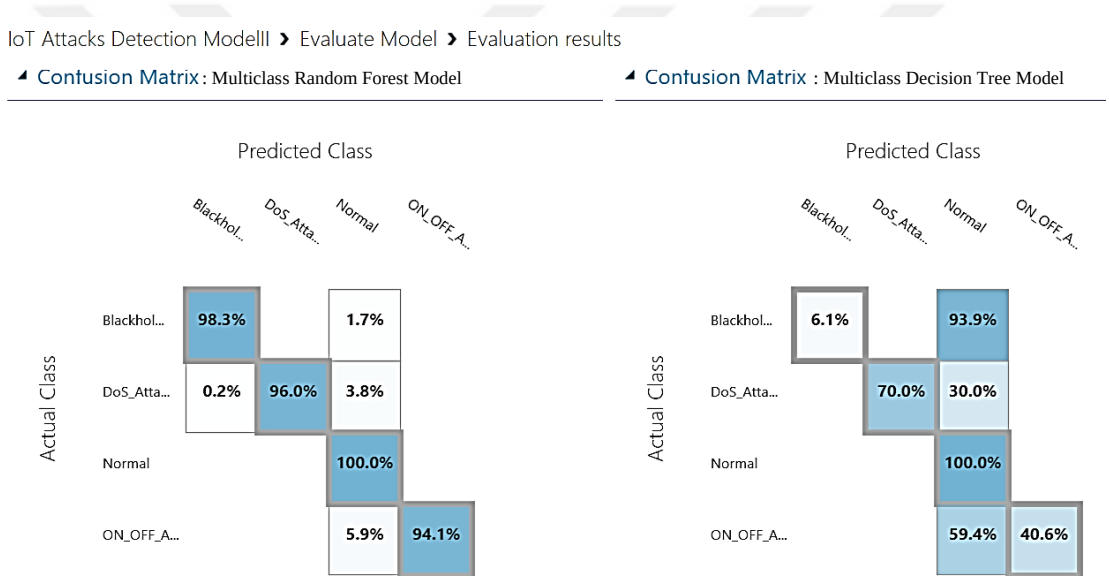


Figure 7.2. Snapshot performance matrix for classification models

7.1. Performance Metrics for Classification Models

This Section illustrates the confusion matrix of classification models in Figure 7.2 in details. A confusion matrix is used to describe the performance of a classification model. It summarizes the testing data and the total of indexes in each matrix must be equal to the sum of the test data samples. Through it, we can calculate the important parameters related to the performance of models like the accuracy, precision, and Recall to evaluation the models.

True positives (TP): The model predicted the nodes have the attacks.

True negatives (TN): The model predicted the nodes don't have the attacks.

False positives (FP): The model predicted the nodes have the attacks, but actually it doesn't have the attacks and this is an error.

False negatives (FN): The model predicted the nodes don't have the attacks, but actually it does have the attacks and this is an error.

Of course, we can observe in the confusion matrix, the number of frequency samples in each class is unbalanced or not close to each other. due to the simulation duration being different from one attack to another as illustrated in Section 4.3. Therefore, the number of samples is different from one attack to another such as the number of samples 9912, 12696, and 25072 from DoS, BHA, and OOA, respectively. The reason for the difference in simulation duration is explained in Section 5.2. As well as the overall number of samples in the normal class is greater than other classes due to the number of normal nodes in the network area being greater than malicious nodes. where only node 8 is the malicious node and the other nodes normal and this lead to unbalance samples in the matrix confusion as in Tables 7.1, 7.2.7.3, 7.4, 7.5,7.6, 7.7, 7.8, 7.9, and 7.10.

7.1.1. The Confusion Matrix for Multiclass Random Forest Model

MRF acronyms for Multiclass Random Forest Model. In the multiclass random forest model, in Table 7.1, the testing data equals the sum of all the index matrices =288+5+1+484+19+7493+73+1172=9535. This number represents 20% of the KoÜ-6LoWPAN-IoT dataset, which contains 47680, while the number of training data is 38142. Tables 7.2, 7.3, 7.4, and 7.5 show in detail the performance matrix for each class.

Table 7.1. Multiclass random decision forest confusion matrix overall classes

(Test samples) N=9535		Predicted Class			
		BHA	DoS	Normal	OOA
Actual Class	BHA	288	0	5	0
	DoS	1	484	19	0
	Normal	0	0	7493	0
	OOA	0	0	73	1172

Table 7.2. BHA class confusion matrix in MRF

(Test samples) N=9535		Predicted Class	
Actual Class	BHA class	BHA=YES	BHA=NO
	BHA=YES	TP=288	FN= 5
	BHA=NO	FP=1	TN=9241

Table 7.3. DoS attack class confusion matrix in MRF

N=9535		Predicted Class	
Actual Class	DoS Class	DoS=YES	DoS=NO
	DoS=YES	TP=484	FN= 20
	DoS=NO	FP=0	TN=9031

$$\text{Accuracy}_{(\text{BHA})} = \frac{\text{TP}_{(288)} + \text{TN}_{(9241)}}{\text{TP}_{(288)} + \text{TN}_{(9241)} + \text{FP}_{(1)} + \text{FN}_{(5)}} = 0.99 \quad (6.1)$$

$$\text{Precision}_{(\text{BHA})} = \frac{\text{TP}_{(288)}}{\text{TP}_{(288)} + \text{FP}_{(1)}} = 0.99 \quad (6.2)$$

$$\text{Recall}_{(\text{BHA})} = \frac{\text{TP}_{(288)}}{\text{TP}_{(288)} + \text{FN}_{(5)}} = 0.98 \quad (6.3)$$

$$\text{Accuracy}_{(\text{DoS})} = \frac{484 + 9031}{484 + 9031 + 0 + 20} = 0.99 \quad (6.4)$$

$$\text{Precision}_{(\text{DoS})} = \frac{484}{484 + 0} = 1 \quad (6.5)$$

$$\text{Recall}_{(\text{DoS})} = \frac{484}{484 + 20} = 0.96 \quad (6.6)$$

Table 7.4. Normal class confusion matrix in MRF

N=9535		Predicted Class	
Actual Class	Normal Class	Normal=YES	Normal=NO
	Normal=YES	TP=7493	FN= 0
	Normal=NO	FP=97	TN=1944

$$\text{Accuracy}_{(Normal)} = \frac{7493+1944}{7493+1944+97+0} = 0.98 \quad (6.7)$$

$$\text{Precision}_{(Normal)} = \frac{7493}{7493+97} = 0.98 \quad (6.8)$$

$$\text{Recall}_{(Normal)} = \frac{7493}{7493+0} = 1 \quad (6.9)$$

Table 7.5. OOA class confusion matrix in MRF

N=9535		Predicted Class	
Actual Class	OOA Class	OOA=YES	OOA=NO
	OOA=YES	TP=1172	FN=73
	OOA=NO	FP=0	TN=8290

$$\text{Accuracy}_{(OOA)} = \frac{1172+8290}{1172+8290+0+73} = 0.99 \quad (6.10)$$

$$\text{Precision}_{(OOA)} = \frac{1172}{1172+0} = 1 \quad (6.11)$$

$$\text{Recall}_{(OOA)} = \frac{1172}{1172+73} = 0.94 \quad (6.12)$$

The Overall Accuracy for classes

$$=\frac{\text{TP }_{(\text{DoS})}+\text{TP }_{(\text{OOA})}+\text{TP }_{(Normal)}+\text{TP }_{(\text{BHA})}}{\sum n}=\frac{288+484+7493+1172}{288+5+484+19+1+7493+73+1172}= \quad (6.13)$$

0.989

The Average Accuracy for classes

$$=\frac{\text{Accuracy }_{(\text{DoS})}+\text{Accuracy }_{(\text{OOA})}+\text{Accuracy }_{(Normal)}+\text{Accuracy }_{(\text{BHA})}}{4}= \quad (6.14)$$

$$\frac{0.99+0.99+0.98+0.99}{4}=0.98$$

The Average Precision for classes

$$=\frac{\text{Precision }_{(\text{DoS})}+\text{Precision }_{(\text{OOA})}+\text{Precision }_{(Normal)}+\text{Precision }_{(\text{BHA})}}{4}= \quad (6.15)$$

$$\frac{1+1+0.98+0.99}{4}=0.99$$

The Average Recall for classes

$$=\frac{\text{Recall }_{(\text{DoS})}+\text{Recall }_{(\text{OOA})}+\text{Recall }_{(Normal)}+\text{Recall }_{(\text{BHA})}}{4}=\frac{0.96+0.94+1+0.98}{4}= \quad (6.16)$$

0.97

7.1.2. The Confusion Matrix for Multiclass Decision Tree Model

MDT acronyms for Multiclass Decision Tree Model. In the multiclass decision tree model, in Table 7.6, the testing data equals the sum of all the index matrices =18+5+275+353+151+7493+740+505=9535. This number represents 20% of the KoÜ-6LoWPAN-IoT dataset, which contains 47680, while the number of training data is 38142. Tables 7.7, 7.8, 7.9, and 7.10 show in detail the performance matrix for each class in multiclass decision tree confusion matrix.

Table 7.6. Multiclass decision tree confusion matrix overall classes

N=9535		Predicted Classes			
		BHA	DoS	Normal	OOA
Actual Class	BHA	18	0	275	0
	DoS	0	353	151	0
	Normal	0	0	7493	0
	OOA	0	0	740	505

Table 7.7. BHA class confusion matrix in MDT

N=9535		Predicted Classes	
Actual Class	BHA class	BHA=YES	BHA=NO
	BHA=YES	TP=18	FN= 275
	BHA=NO	FP=0	TN=9242

$$\text{Accuracy}_{(BHA)} = \frac{TP_{(18)} + TN_{(9242)}}{TP_{(18)} + TN_{(9242)} + FP_{(0)} + FN_{(275)}} = 0.97 \quad (6.17)$$

$$\text{Precision}_{(BHA)} = \frac{TP_{(18)}}{TP_{(18)} + FP_{(0)}} = 1 \quad (6.18)$$

$$\text{Recall}_{(BHA)} = \frac{TP_{(18)}}{TP_{(18)} + FN_{(275)}} = 0.06 \quad (6.19)$$

Table 7.8. DoS attack class confusion matrix in MDT

N=9535		Predicted Classes	
Actual Class	DoS class	DoS=YES	DoS=NO
	DoS=YES	TP=353	FN= 151
	DoS=NO	FP=0	TN=9031

Table 7.9. Normal class confusion matrix in MDT

N=9535		Predicted Classes	
Actual Class	Normal Class	Normal=YES	Normal=NO
	Normal=YES	TP=7493	FN= 0
	Normal=NO	FP=1166	TN=876

$$\text{Accuracy}_{(DoS)} = \frac{353 + 9031}{353 + 9031 + 0 + 151} = 0.98 \quad (6.20)$$

$$\text{Precision}_{(DoS)} = \frac{353}{353 + 0} = 1 \quad (6.21)$$

$$\text{Recall}_{(DoS)} = \frac{353}{353 + 151} = 0.70 \quad (6.22)$$

$$\text{Accuracy}_{(Normal)} = \frac{7493 + 876}{7493 + 876 + 1166 + 0} = 0.87 \quad (6.23)$$

$$\text{Precision}_{(Normal)} = \frac{7493}{7493 + 1166} = 0.86 \quad (6.24)$$

$$\text{Recall}_{(Normal)} = \frac{7493}{7493 + 0} = 1 \quad (6.25)$$

Table 7.10. OOA class confusion matrix in MDT

N=9535		Predicted Classes	
Actual Class	OOA Class	OOA=YES	OOA=NO
	OOA=YES	TP=505	FN=740
	OOA=NO	FP=0	TN=8290

$$\text{Accuracy}_{(OOA)} = \frac{505 + 8290}{505 + 8290 + 0 + 740} = 0.92 \quad (6.26)$$

$$\text{Precision}_{(OOA)} = \frac{505}{505 + 0} = 1 \quad (6.27)$$

$$\text{Recall}_{(OOA)} = \frac{505}{505 + 740} = 0.40 \quad (6.28)$$

The Overall Accuracy for classes

$$= \frac{\text{TP}_{(\text{DoS})} + \text{TP}_{(\text{OOA})} + \text{TP}_{(Normal)} + \text{TP}_{(\text{BHA})}}{\sum n} = \frac{18 + 353 + 7493 + 505}{18 + 275 + 353 + 151 + 7493 + 740 + 505} = 0.877 \quad (6.29)$$

The Average Accuracy for classes

$$= \frac{\text{Accuracy}_{(\text{DoS})} + \text{Accuracy}_{(\text{OOA})} + \text{Accuracy}_{(Normal)} + \text{Accuracy}_{(\text{BHA})}}{4} = \frac{0.98 + 0.92 + 0.87 + 0.97}{4} = 0.935 \quad (6.30)$$

The Average Precision for classes

$$= \frac{\text{Precision}_{(\text{DoS})} + \text{Precision}_{(\text{OOA})} + \text{Precision}_{(Normal)} + \text{Precision}_{(\text{BHA})}}{4} = \frac{1 + 1 + 0.86 + 1}{4} = 0.965 \quad (6.31)$$

The Average Recall for classes

$$= \frac{\text{Recall}_{(\text{DoS})} + \text{Recall}_{(\text{OOA})} + \text{Recall}_{(Normal)} + \text{Recall}_{(\text{BHA})}}{4} = \frac{0.70+0.40+1+0.06}{4} = 0.54 \quad (6.32)$$

7.2. Performance Metrics for Regression Models

The Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), Relative Squared Error (RSE), Coefficient of Determination(r-squared) are used in regression models to measure the performance of a predictive model and measure the accuracy. Therefore, the better predictions model has an MSE, RMSE, RAE, and RSE value closer to zero. as well as the Coefficient of Determination is better when the its value closer to one. Figure 7.3 illustrate a snapshot of matrix performance for the decision random forest and boosted decision tree for the regression model.

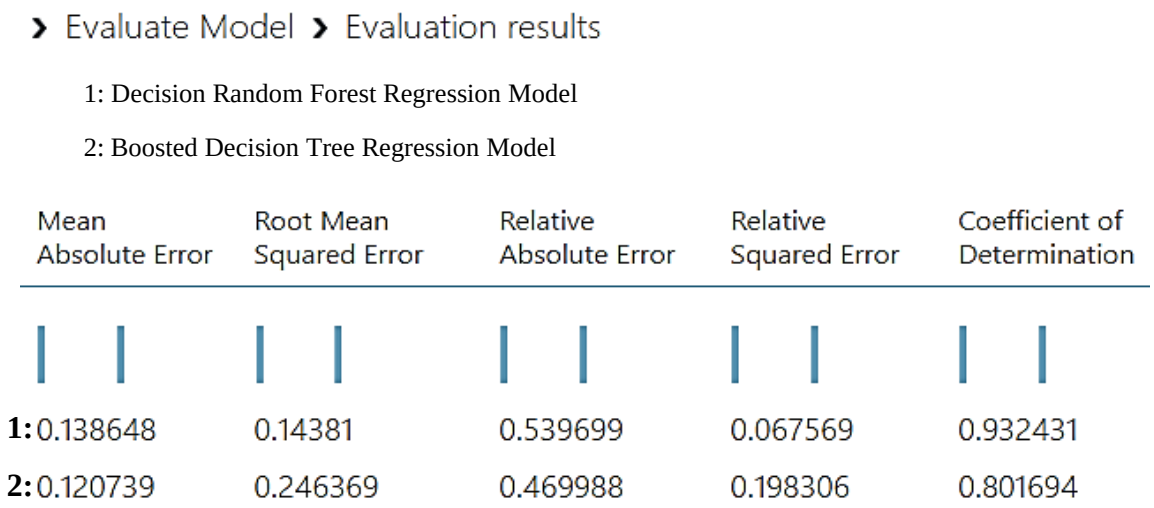


Figure 7.3. Snapshot of performance matrix for regression models

7.2.1. Performance Matrix for Decision Random Forest Model

Table 7.11 shows in detail the performance matrix for the decision random forest model. where the r-squared (coefficient of determination) was approximately 0.93. This value is better when it is close to one because it describes the strength of the relationship of input dependency to output.

Table 7.11. Performance matrix for decision forest regression model

Model Evaluation	MAE	RMSE	RAE	RSE	r-squared
Decision Forest Regression Model	0.138648	0.14381	0.539699	0.067569	0.932431

7.2.2. Performance Matrix for Boosted Decision Tree Model

In Table 7.12 we can see in detail the performance matrix for a boosted decision tree model. we observed the r-squared was approximately 0.80. This value of CD is better when it is close to one because it describes the strength of the relationship of input dependency to output. Increasing the CD prediction value means increasing the accuracy of the model. As well as other factors like MAE, RMSE, RAE, and RSE are better when them are close to zero because these factors indicated an error in the model. and decreasing these values means increasing the accuracy of the model.

Table 7.12. Performance matrix for boosted decision tree model

Model Evaluation	MAE	RMSE	RAE	RSE	r-squared
Boosted Decision Tree Model	0.120739	0.246369	0.469988	0.198306	0.801694

As a result, as shown in Figure 7.4, the multiclass random decision forest ML-based model obtained 98.9% overall accuracy in identifying IoT attacks for the KoÜ-6LoWPAN-IoT dataset IoT features-based, compared to 87.7%, 93.2%, and 87.1% for decision tree, decision random forest tree regression, and boosted decision tree regression, respectively. As a consequence, multiclass random decision forest produces fair results when compared to other algorithms, but this does not exclude the development of lightweight custom ML algorithms for future IoT challenges

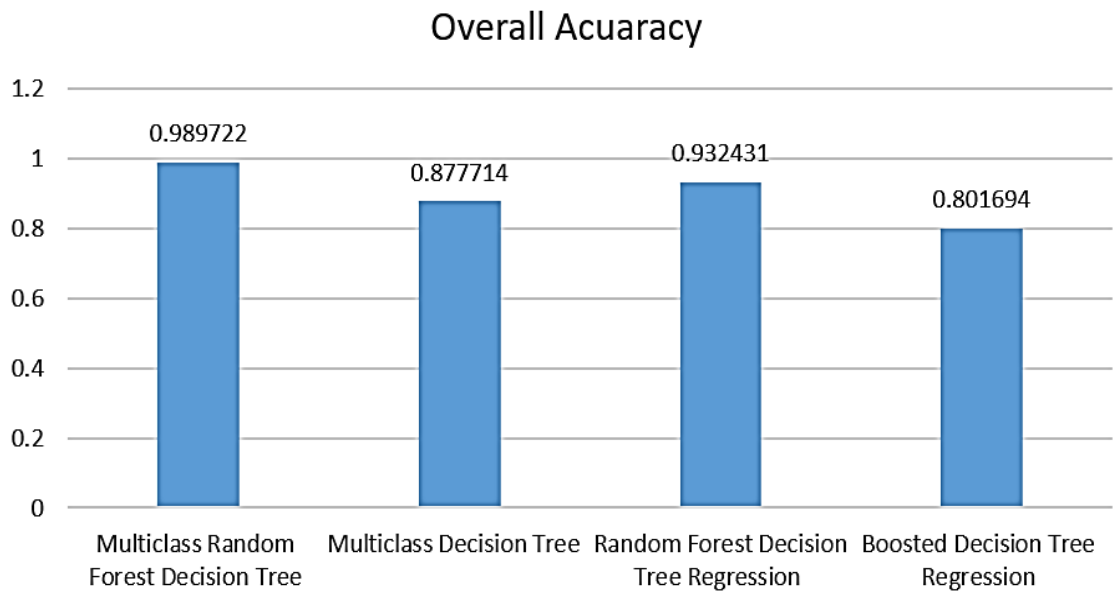


Figure 7.4. The overall accuracy of decision tree-based models

A random sample was selected from each attack and tested using external tool which is postman tool on the multiclass decision random decision forest model as in Figures 7.5, 7.6, 7.7, and 7.8. The highest predictive value of the new data tested in Figure 7.5 is 78.8%, therefore the score label is DoS attack. While the predictive value in Figure 7.6 is 90.3%, the score label is BHA attack. In Figures 7.7 and 7.8, the highest scored probabilities are 52.5% and 94%, thus the score label is OOA and normal, respectively. Finaly,the proposed model has the ability to detect and prevent attacks in addition to identifying the source and destination of the attacks in a practical and effective manner.



Figure 7.5. New predictive value of DoS attacks

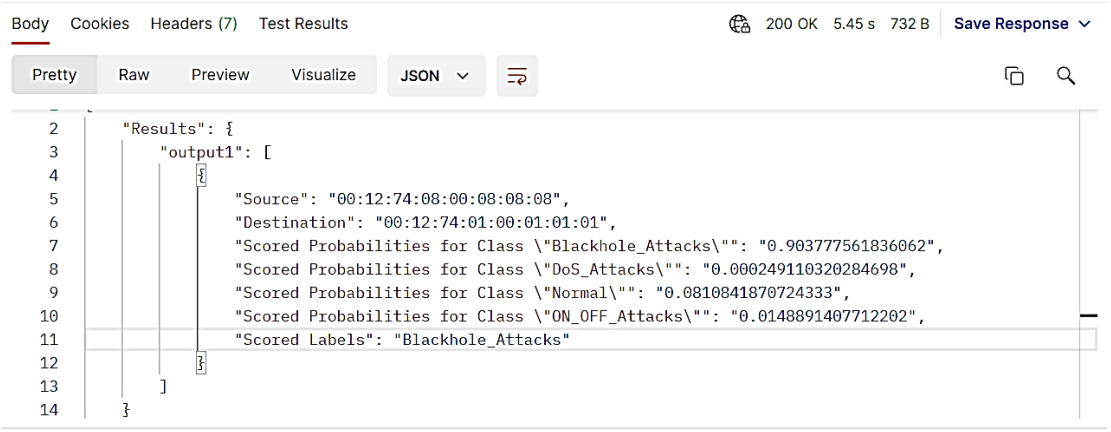


Figure 7.6. New predictive value of BHA

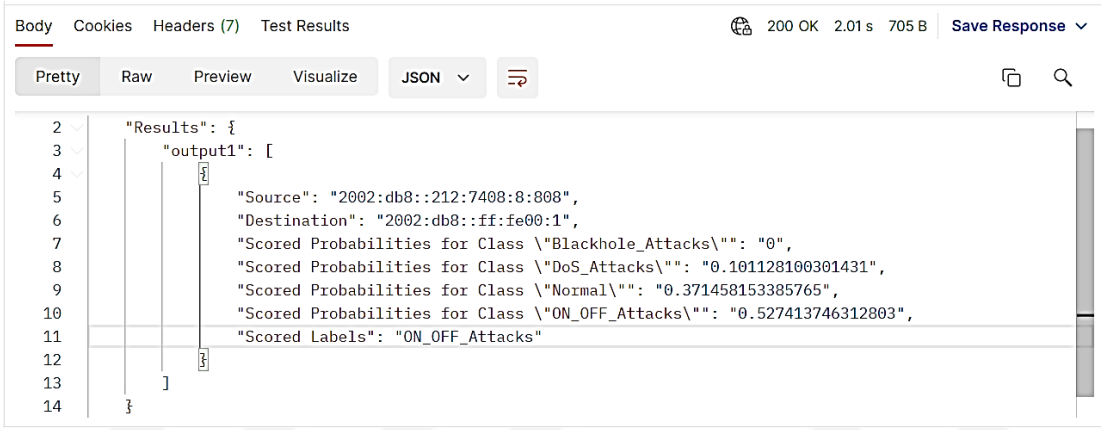


Figure 7.7. New predictive value of OOA

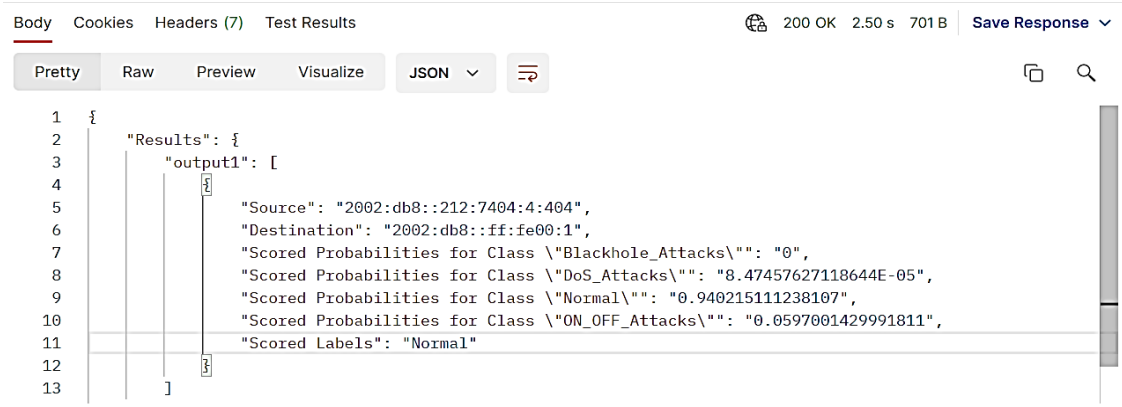


Figure 7.8. New predictive value of Normal

8. CONCLUSION

Due to heterogeneity, dynamic ecosystems, and connection to an unsecured and trusted world, the IoT ecosystems are susceptible to a variety of attacks and are considered challenging. Therefore, employing the latest technology to address security issues in the IoT has become extremely important in order to enhance security in smart systems. However, enabling ML-based techniques to address IoT security issues is a strong point.

This thesis discusses the implementation of attacks, their analysis features, and the approach to detecting attacks and deployment models in the IoT network to address the security issues in IoT ecosystem. Therefore, we have developed and implemented three attacks in the IoT ecosystem via the Cooja simulator. Not just for myself, but for many other researchers, the open-source Contiki OS was a challenging point to carry out attacks. In this context, we selected the attacks based on their degree of difficulty in implementation, detection, popularity, spread, and severity. The selected attacks are arranged as follows: DoS, BHA, and OOA attacks. The selected types of attacks based on certain criteria give the models more accuracy due to the variety of attacks and give us to examine the model very well. The technology on which the attacks are carried out is an important factor. Therefore, we configured all scenarios on 6LoWPAN stacks for the IoT, consisting of UDP, IPv6, ICMP, RPL, 6LoWPAN, and IEEE 802.15.4. The 6LoWPAN devices achieve interoperability, which implies that apps do not need to know the limits of the physical connections that transport their packets unlike their peers like Zigbee. 6LoWPAN devices enable communication with any wireless 802.15.4 device over any other IP network connection (e.g., Ethernet or Wi-Fi).

It was necessary to analyze the attacks and extract the criteria before proposing the approaches to detect attacks. After developing the attacks and implementing them and tracing the behaviors between the normal and malicious nodes, therefore, we analyzed the attacks to know the parameters/features that are used as an evaluation for analyzing attacks in the scenes as well as understanding the distinguished parameters of normal and malicious activities between scenarios. furthermore, characteristics or features affected by the attacks are an obvious criterion and input for Intrusion Detection systems (IDS) or features of the datasets that can be employed in Machine Learning (ML) models. After analyzing the attacks, we were able to determine the criteria for evaluating the attacks

that are affected when the attacks are executed, which are radio messages and power consumption. The reason is that when implementing any scenario and following the behavior of the malicious node, we noted that the power consumption of the harmful node is very high compared to normal nodes, as well as the features of radio message (6loWPAN protocol stack) in malicious activities are affected due to the change and update matrix/variables in each process as well as these features are always available on each device. The analysis of attacks gave us an indicator to explore and extract the features affected by the attacks to produce the KoÜ-6LoWPAN-IoT dataset, which was generated using Cooja to be employed in ML-Models. The dataset was generated based on the general characteristics that might be affected by any attack power consumption and radio messages. The KoÜ-6LoWPAN-IoT dataset IoT-based features contain a mix of data from malicious nodes and normal nodes in the IoT ecosystem. It contains many numbers of samples of attacks collected from DoS, BHA, and OOA attacks, which total 47680 samples of observations, 84 features from radio messages, and 12 from the power consumption features.

As for attack detection methods, we discarded the IDS proposed for detecting certain attacks. This may be effective to detect a special or certain attack, but implementing an IDS mechanism for each attack, especially with advances in technology and increasing attack types, may be inefficient. It may consume device resources as well as prevent it from being extended to detect new attacks. Therefore, we enabled ML in IoT for learning and analyzing the behaviors of IoT devices/objects, and systems based on prior information and experiences may allow the IoT ecosystem to effectively manage the unexpected deterioration and detect attacks that are frequently caused by anomalous conditions. As well as the machine learning-based approaches can classify more than one attack. Hence, enabling machine learning to detect more than one attack is better than designing and developing IDS for each attack. Therefore, we have proposed an approach to detect IoT attacks that depend on ML-based approaches. The ML-based approaches depend on decision tree-based models that have proven their efficiency in manipulating, examining, and classifying malicious activity of the IoT features generated via the Cooja simulator. The ML decision tree based includes RF, DT, and BDT which has the structure of a tree and takes its decisions based on the output of the tree. In the context of this work, the multiclass Random Decision Forest ML-based model achieved 98.9% overall

accuracy in detecting IoT attacks for the KoÜ-6LoWPAN-IoT dataset compared to the decision tree, decision forest tree regression, and boosted decision tree regression, which achieved 87.7%, 93.2%, and 87.1%, respectively. Therefore, the multiclass random forest has less computational complexity with high accuracy, so it might be a fair model and a lightweight algorithm to detect attacks in the IoT. In addition, random samples were tested on each attack in the model that achieved the highest accuracy, which was the multiclass random decision forest. As the orientation of designing the lightweight ML custom algorithm to solve the IoT security problem. It is a matter of interest and tends to be on the minds of developers.

We demonstrated the lightweight and optimal ML models based on time complexity and accuracy as well as the deployment of models in the network. Therefore, the decision tree-based algorithm has less computational complexity, and the model's accuracy ranges between high and moderate. So, this type of model achieves lightweight models and optimal models. While other models like KNN, SVM, and ANN have more computational complexity, their accuracy ranges between high and moderate. So, this type of model achieves heavyweight models and optimal models. Therefore, finding the lightweight and heavyweight models gave us a decision to deploy models that will detect attacks and which may be in the cloud, fog, or edge computing. Therefore, the decision tree-based model deployment will be suitable for edge computing, unlike models that require more computational complexity. The models' deployment will be suitable for cloud computing or fog computing.

Developing a physical network to implement attacks and gathering data to provide an actual dataset as input for training and testing our ML models and designing custom IDS and examining them on the actual dataset as well as comparing them may be the next step. In the future related to this work, previous work will be carried out in an actual environment to implement attacks and generate an actual dataset and test it on various ML-based approaches. We will also develop and design a customized Intelligent Adaptive Tiny-Intrusion Detection System (IAT-IDS) to detect various attacks in the IoT ecosystem that work on different devices and techniques, especially tiny devices and, test it on an actual dataset. The IAT-IDS will achieve the interoperability imposed by the limitations of the standards on different devices and technologies in their physical and

data link layers (Ex: 802.15.4 PHY and MAC). Once the IIAT-IDS identifiers are downloaded, the device features and characteristics are automatically identified. Automatically, the IIAT-IDS will initialize a layer called the "detection layer" to detect attacks and protect the devices from any risks.

REFERENCES

Abdulkareem, N. M., Zeebaree, S. R., M. Sadeeq, M. A., Ahmed, Di. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud Computing Issues, Challenges and Opportunities: A Review. *Qubahan Academic Journal*, 1–7. https://doi.org/10.48161/issn.2709-8206

Abughazaleh, N., Bin, R., Btish, M., & M., H. (2020). DoS Attacks in IoT Systems and Proposed Solutions. *International Journal of Computer Applications*, *176*(33), 16–19. https://doi.org/10.5120/ijca2020920397

Ahmed, Firoz, A., & Young-Bae, K. (2016). Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks. *Security and Communication Networks*, *9*(18), 5143–5154. https://doi.org/10.1002/sec.1684

Airehrour David, Gutierrez Jairo, & Kumar Ray Sayan. (2017). Securing RPL routing protocol from blackhole attacks using a trust-based mechanism. *26th International Telecommunication Networks and Applications Conference, ITNAC 2016*, 115–120. https://doi.org/10.1109/ATNAC.2016.7878793

Al-Akhras, M., Alawairdhi, M., Alawairdhi, A., & Atawneh, S. (2020). Using Machine Learning To Build A Classification Model For Iot Networks To Detect Attack Signatures. *International Journal of Computer Networks and Communications*, *12*(6), 99–116. https://doi.org/10.5121/ijcnc.2020.12607

Al-Kaseem, B. R., & Al-Raweshidyhamed, H. S. (2017). SD-NFV as an Energy Efficient Approach for M2M Networks Using Cloud-Based 6LoWPAN Testbed. *IEEE Internet of Things Journal*, *4*(5), 1787–1797. https://doi.org/10.1109/JIOT.2017.2704921

ALAOUI ISMAILI, I., Azyat, A., Raissouni, N., Ben Achhab, N., Chahboun, A., & Lahraoua, M. (2019). *Comparative Study of ZigBee and 6LoWPAN Protocols: Review*. 2–10. https://doi.org/10.4108/eai.24-4-2019.2284215

Ali, S., Khan, M. A., Ahmad, J., Malik, A. W., & Ur Rehman, A. (2018). Detection and prevention of Black Hole Attacks in IOT & WSN. *2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018*, 217–226. https://doi.org/10.1109/FMEC.2018.8364068

Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., & Ming, H. (2019). AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019*, 305–310. https://doi.org/10.1109/CCWC.2019.8666450

Bagui, S., Wang, X., & Bagui, S. (2021). Machine Learning Based Intrusion Detection for IoT Botnet. *International Journal of Machine Learning and Computing*, *11*(6), 399–406. https://doi.org/10.18178/ijmlc.2021.11.6.1068

Bayılmış, C., Ebleme, M. A., Çavuşoğlu, Ü., Küçük, K., & Sevin, A. (2022). A survey on

communication protocols and performance evaluations for Internet of Things. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2022.03.013

Bedi, P., Mewada, S., Vatti, R. A., Singh, C., Dhindsa, K. S., Ponnusamy, M., & Sikarwar, R. (2021). Detection of attacks in IoT sensors networks using machine learning algorithm. *Microprocessors and Microsystems*, *82*. https://doi.org/10.1016/j.micpro.2020.103814

Bediya, A. K., & Kumar, R. (2020). Real time DDoS intrusion detection and monitoring framework in 6LoWPAN for internet of things. *2020 IEEE International Conference on Computing, Power and Communication Technologies, GUCON 2020*, 824–828. https://doi.org/10.1109/GUCON48875.2020.9231139

Charles, A. S. J., & Kalavathi, P. (2018). QoS Measurement of RPL using Cooja Simulator and Wireshark Network Analyser View project QoS Measurement of RPL using Cooja Simulator and Wireshark Network Analyser. *Article in INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*. https://www.researchgate.net/publication/325593751

Chen, Y., Chanet, J., Hou, K., Chen, Y., Chanet, J.-P., & Mean HOU, K. (2012). *RPL Routing Protocol a case study: Precision agriculture*. https://hal.archives-ouvertes.fr/hal-00681319

Choudhary, D. (2019). Security Challenges and Countermeasures for the Heterogeneity of IoT Applications. *Journal of Autonomous Intelligence*, *1*(2), 16. https://doi.org/10.32629/jai.v1i2.25

Choudhary, S., & Kesswani, N. (2019). A survey: Intrusion detection techniques for internet of things. *International Journal of Information Security and Privacy*, *13*(1), 86–105. https://doi.org/10.4018/IJISP.2019010107

Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B., & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. *Sensors (Switzerland)*, *21*(2), 1–32. https://doi.org/10.3390/s21020446

Contiki tutorials. (2016). *Contiki 6LoWPAN Quick Guide Contiki on STM32 Nucleo plugged with Sub-1 GHz RF expansion board (X-NUCLEO-IDS01A4, X-NUCLEO-IDS01A5)*. https://github.com/contiki-os/contiki

URL-1: https://iotbyhvm.ooo/contiki-the-open-source-os-for iot/,(Last Access: 10January 2022)

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761–768. https://doi.org/10.1016/j.future.2017.08.043

Djuitcheu, H., Debes, M., Aumuller, M., & Seitz, J. (2022). Recent review of Distributed Denial of Service Attacks in the Internet of Things. *5th Conference on Cloud and Internet of Things, CIoT 2022*, *May*, 32–39.

https://doi.org/10.1109/CIoT53061.2022.9766655

Dobbins, R., Migault, D., Moskowitz, R., Teague, N., Xia, L., & Nishizuka, K. (2021). *RFC 8903: Use Cases for DDoS Open Threat Signaling*. https://www.rfc-editor.org/info/rfc8903

Dunkels, A., Eriksson, J., Finne, N., & Tsiftes, N. (2011). *Powertrace: Network-level Power Profiling for Low-power Wireless Networks*.

Farea, A. H., & Küçük, K. (2021). Multidisciplinary Studies and Innovative Technologies Enhancement Trust Management in IoT to Detect ON-OFF Attacks with Cooja. *Dergipark*, *5*, 123–128.

Farea, A. H., & Küçük, K. (2022). Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja. *EAI Endorsed Transactions on Internet of Things*, *7*(28), 1–12. https://doi.org/10.4108/eetiot.v7i28.324

Gracia Osorio, O. A., Reyes Dazai, B. S., & Salcedo, O. J. (2016). Comparative study of performance for 804.15.4 ZigBee and 6LoWPAN protocols. *CEUR Workshop Proceedings*, *1548*, 59–71.

Hajjaji, Y., Boulila, W., Farah, I. R., Romdhani, I., & Hussain, A. (2021). Big data and IoT-based applications in smart environments: A systematic review. In *Computer Science Review*. https://doi.org/10.1016/j.cosrev.2020.100318

Hasan, M., Milon Islam, M., Ishrak Islam Zarif, M., & Hashem, M. (2019). *Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches*. https://doi.org/10.1016/j.iot.2019.10

Hennebert, C., & Dos Santos, J. (2014). Security Protocols and Privacy Issues into 6LoW-PAN Stack: A Synthesis. *IEEE Internet of Things Journal*, *1*(5), 384–398. https://doi.org/10.1109/JIOT.2014.2359538ï

Ioulianou, P. P., & Vassilakis, V. G. (2020). Denial-of-service attacks and countermeasures in the RPL-based internet of things. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11980 LNCS*(February), 374–390. https://doi.org/10.1007/978-3-030-42048-2_24

Johnsen, F. T., Zielinski, Z., Wrona, K., Suri, N., Fuchs, C., Pradhan, M., Furtak, J., Vasilache, B., Pellegrini, V., Dyk, M., Marks, M., & Krzyszton, M. (2018). Application of IoT in military operations in a smart city. *2018 International Conference on Military Communications and Information Systems, ICMCIS 2018*, *December*, 1–8. https://doi.org/10.1109/ICMCIS.2018.8398690

Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018). A Survey on Anomaly Based Host Intrusion Detection System. *Journal of Physics: Conference Series*, *1000*(1). https://doi.org/10.1088/1742-6596/1000/1/012049

Jyothisree, J., & Sreekanth, S. (2019). *Attacks in RPL and Detection Technique used for*

Internet of Things.

Kale, & Bhosale. (2021). *Detection of blackhole attack in distributed wireless sensor networks*. *9*(3), 172–175.

Kasinathan, P., Pastrone, C., Spirito, M. A., & Vinkovits, M. (2013). Denial-of-Service detection in 6LoWPAN based Internet of Things. *International Conference on Wireless and Mobile Computing, Networking and Communications*, 600–607. https://doi.org/10.1109/WiMOB.2013.6673419

Kim, E., Kaspar, D., & Vasseur, J. (2012). Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). *Rfc*, 1–28.

Labraoui, N., Gueroui, M., & Sekhri, L. (2018). *On-Off Attacks Mitigation against Trust Systems in Wireless Sensor Networks To cite this version : HAL Id : hal-01789947 On-Off Attacks Mitigation against Trust Systems in Wireless Sensor Networks*. 406–415.

URL-2: https://www.kaggle.com/datasets/francoisxa/ds2ostraffictraces , (Last Access: 10 May 2022).

URL-3: https://www.unb.ca/cic/datasets/index.html , (Last Access: 17 May 2022).

URL-4: https://research.unsw.edu.au/projects/unsw-nb15-dataset, (Last Access: 22 May 2022).

URL-5: https://research.unsw.edu.au/projects/bot-iot-dataset, (Last Access: 29 May 2022).

Laghari, A. A., Wu, K., Laghari, R. A., Ali, M., & Khan, A. A. (2022). A Review and State of Art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*, *29*(3), 1395–1413. https://doi.org/10.1007/s11831-021-09622-6

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008

Mekala, M. S., & Viswanathan, P. (2017). A Survey: Smart agriculture IoT with cloud computing. *2017 International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2017*, *2017-Janua*(August 2017), 1–7. https://doi.org/10.1109/ICMDCS.2017.8211551

Mendoza, C. V. L., & Kleinschmidt, J. H. (2015). Mitigating on-off attacks in the internet of things using a distributed trust management scheme. *International Journal of Distributed Sensor Networks*, *2015*. https://doi.org/10.1155/2015/859731

Minh Dang, L., Piran, M. J., Han, D., Min, K., & Moon, H. (2019). A survey on internet of things and cloud computing for healthcare. *Electronics (Switzerland)*, *8*(7), 1–49. https://doi.org/10.3390/electronics8070768

Moradi, F., Sedaghatbaf, A., Asadollah, S., & Sirjani, M. (2019). On-Off Attack on a Blockchain-based IoT System. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *2019-Septe*, 1768–1773. https://doi.org/10.1109/ETFA.2019.8868238

Nasution, A., Suryani, V., & Wardana, A. (2020). *IoT Object Security towards On-off Attack Using Trustworthiness Management*.

Neerugatti, V., & Reddy, A. R. M. (2019). Detection and prevention of black hole attack in RPL protocol based on the threshold value of nodes in the internet of things networks. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9 Special Issue 3), 325–329. https://doi.org/10.35940/ijitee.I3060.0789S319

Patel, H. B., & Jinwala, D. C. (2021). 6MID: Mircochain based intrusion detection for 6LoWPAN based IoT networks. *Procedia Computer Science*, *184*, 929–934. https://doi.org/10.1016/j.procs.2021.04.023

Perrone, L. F., & Nelson, S. C. (2006). A study of on-off attack models for wireless ad hoc networks. *2006 1st Workshop on Operator-Assisted (Wireless-Mesh) Community Networks, OpComm 2006*. https://doi.org/10.1109/WOACN.2006.337180

Pflanzner, T., & Kertesz, A. (2018). A Taxonomy and Survey of IoT Cloud Applications. *EAI Endorsed Transactions on Internet of Things*, *3*(12), 154391. https://doi.org/10.4108/eai.6-4-2018.154391

Ping, H., Wang, J., Ma, Z., & Du, Y. (2018). Mini-review of application of IoT technology in monitoring agricultural products quality and safety. *International Journal of Agricultural and Biological Engineering*, *11*(5), 35–45. https://doi.org/10.25165/j.ijabe.20181105.3092

Ramakrishnan, A. K., Preuveneers, D., & Berbers, Y. (2014). Enabling self-learning in dynamic and open Iot environments. *Procedia Computer Science*, *32*, 207–214. https://doi.org/10.1016/j.procs.2014.05.416

Rani, D., & Kaushal, N. C. (2020). Supervised Machine Learning Based Network Intrusion Detection System for Internet of Things. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*. https://doi.org/10.1109/ICCCNT49239.2020.9225340

URL-6: https://www.kaggle.com/general/263127,Last access 17-7-2022, (Last Access: 18 June 2022).

URL-7:https://medium.com/analytics-vidhya/time-complexity -of-ml-models-4ec39fad2770, (Last Access: 7 July 2022).

URL-8:https://medium.com/analytics-vidhya/computational -complexity-of-ml-algorithms-1bdc88af1c7a, (Last Access: 12 July 2022).

Rghioui, A., Bouhorma, M., & Benslimane, A. (2013). Analytical study of security aspects

in 6LoWPAN networks. *2013 5th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2013*. https://doi.org/10.1109/ICT4M.2013.6518912

Sahay Rashmi, Geethakumari G., Mitra Barsha, & Thejas V. (2018). *Exponential Smoothing based Approach for Detection of Blackhole Attacks in IoT*.

Seliem, M., & Elgazzar, K. (2019, January 18). IoTeWay: A Secure Framework Architecture for 6LoWPAN Based IoT Applications. *2018 IEEE Global Conference on Internet of Things, GCIoT 2018*. https://doi.org/10.1109/GCIoT.2018.8620137

Shah, K., & Narmavala, Z. (2018). A Survey on Green Internet of Things. *14th International Conference on Information Processing: Internet of Things, ICInPro 2018 - Proceedings*, *17*(4), 2347–2376. https://doi.org/10.1109/ICINPRO43533.2018.9096789

Sicari, S., Rizzardi, A., Miorandi, D., & Coen-Porisini, A. (2018). REATO: REActing TO Denial of Service attacks in the Internet of Things. *Computer Networks*, *137*(March), 37–48. https://doi.org/10.1016/j.comnet.2018.03.020

Sikarwar, R., Yadav, P., & Dubey, A. (2020). *9 th IEEE International Conference on Communication Systems and Network Technologies A Survey on IOT enabled cloud platforms*. 120–124. https://doi.org/10.1109/CSNT.2020.23

Suryani, V., Sulistyo, S., & Widyawan. (2018). The Detection of On-Off Attacks for the Internet of Things Objects. *Proceedings - 2018 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC 2018*, 1–5. https://doi.org/10.1109/ICCEREC.2018.8712098

Tanveer, M., Abbas, G., Abbas, Z. H., Waqas, M., Muhammad, F., & Kim, S. (2020). S6AE: Securing 6lowpan using authenticated encryption scheme. *Sensors (Switzerland)*, *20*(9), 1–23. https://doi.org/10.3390/s20092707

Tanwar, S. (2013). Threats & Security Issues in Ad hoc network: A Survey Report. *International Journal of Soft Computing and Engineering*, *26*, 2231–2307.

Thamaraiselvi, D. R., & Anitha Selva Mary, S. (2020). Attack and Anomaly Detection in IoT Networks using Machine Learning. *International Journal of Computer Science and Mobile Computing*, *9*(10), 95–103. https://doi.org/10.47760/ijcsmc.2020.v09i10.012

Verma, A., & Ranga, V. (2020). Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review. *IEEE Sensors Journal*, *20*(11), 5666–5690. https://doi.org/10.1109/JSEN.2020.2973677

Zhang, H., Yao, D., Ramakrishnan, N., & Zhang, Z. (2016). Causality reasoning about network events for detecting stealthy malware activities. *Computers and Security*, *58*, 180–198. https://doi.org/10.1016/j.cose.2016.01.002

URL-9: https://studio.azureml.net/, (Last Access: 30 July 2022). (Last Access: 18 August 2022).

APPENDICES

Appendix-A

In the appendix section, we displayed and illustrated the code structure of malicious nodes and normal nodes in three scenarios. This appendix doesn't include all the statements of code. The open-source file for these scenarios is a file called IoT_Attacks. You can follow the Read_Me file to open these scenarios.

```
/*********************************************************************/
/*DoS ATTACKS*/
/*********************************************************************/
DEFINITIONS:
// Set the sending time for malicious nodes to one second.
# define Malicious_send_period_time 1 sec
# SEND_INTERVAL
// Set the sending time to 1 minute for normal nodes.
# define Normal_send_period_time 1 min
#SEND_INTERVAL_1
START PROCESS:
While (true) {
//Define timer for normal node
Timer1 (&periodic, SEND_INTERVAL);
//Define timer for malicious node
Timer2(&periodic, SEND_INTERVAL_1);
//The malicious node sends a packet every 1 second and then calls the send_packet function.
// We denoted the malicious node by its ID=8
 if (Node_ID==8) {
 Timer1(Malicious_send_period_time, Send_packet);
 PRINTF ("Malicious node\n");}
 else {
// The normal nodes send a packet every 1 minute and then calls the send packet function.
 Timer2 (Normal_send_period_time_1, Send_packet);
 PRINTF ("Normal node\n");}
```

END PROCESS:

}

Send_packet {

If (Node_ID==8) {

// Malicious Client Request to the Server Every Second (Adverse case)

Client →Server

PRINTF ("DoS attack is active, it sends to server ", Number of Req.);}

else {

// Normal Clients Request to the server every one minute (Normal case)

Clients →Server

PRINTF ("Client is Normal, it sends to server ", Number of Req.);}

}}

/**/

Appendix-B

This appendix includes the structure code of the Blackhole attack in the Cooja simulator.

```
/*********************************************************************/
/*BLACK HOLE ATTACKS*/
/*********************************************************************/
// we denoted the malicious node by its ID=8
// Global variables like uip_len, uip_buf, flags and etc.
// The variables must usually be greater than zero due to the updating and changing of their values during each process
If (Node_ID==8 & Global_variables>=0) {
//Go to the drop statement                    (Adverse case)
Goto drop:
}
else {
// Continue to process and execute normal node (Normal case)
 Continuo to call functions and execute it;
}
//Configured global variables and reset them to zero to drop or discard the packages
drop:
// The drop statement returns these values to zero.
// The malicious node may or may not continue to receive and forward the data packets generated by other nodes.
// The malicious node may not continue to process its generated packet.
Set Global variables=0;
}
/*********************************************************************/
```

Appendix-C

This appendix includes the structure code of the on-off attack in the Cooja simulator.

```
/*********************************************************************/
/*ON OFF ATTACKS */
/*********************************************************************/
DEFINITIONS:
// Define trusted variables like max payload length, time of send and so on.
StructTrusted_Variables (MAX_PAYLOAD_LEN, SEND_TIME, SEND_INTERVAL)
{
# define Trusted_paramaters_of packages
}
// Define untrusted variables like max payload length, time of send and so on.
StructUntrusted_Variables (MAX_PAYLOAD_LEN, SEND_TIME,
SEND_INTERVAL) {
# define Untrusted_ parameters _of packages
}
START PROCESS:
While (true) {
// Calls the send_packet function if the sequence of node 8 is even number (Adverse case
for malicious node 8)
  if ((node_id ==8) &(seq_id % 2 == 0)) {
  Timer1 (Trusted_Variables, Send_Packet);
//Keep the cycle ON as possible
  NETSTACK_MAC.off (0.5);
  PRINTF ("Trusted node\n");
 }
// Calls the send_packet function if the sequence of node 8 is odd number, (Normal case
for malicious node 8)
  else if ((node_id ==8) & (seq_id % 2! =0)) {
// The normal nodes send a packet every 1 minute and then calls the send packet function.
```

```
  Timer2 (Untrusted_Variables, Send_packet);
//Keep the cycle ON as possible
  NETSTACK_MAC.off (0.5);
  PRINTF ("Untrusted node\n"); }
Else {
//Here implements another node normal.   (Normal case)
  Timer1 (Trusted_Variables, Send_packet);
  PRINTF ("Normal node\n");}
```

END PROCESS:

```
Send_packet {
//Even and odd case to achieves inconsistent behaviors alternately.
//Even case use Trusted Variables
If ((node_id ==8) &(seq_id % 2 ==0)) {
Use Trusted Variables;
// Malicious Client Request to the Server (Adverse case)
Client →Server
PRINTF ("OOA attack is active, it sends to server ", Number of Req.)}
else if ((node_id ==8) & (seq_id % 2! =0)){
// Odd case use Untrusted Variables;
Use Untrusted Variables;
// Malicious Clients Request to the server (Normal case)
Clients →Server
PRINTF ("OOA attack is active, it sends to server ", Number of Req.}
else {
Use Trusted Variables;
// Normal Clients Request to the Server (Normal case)
Client →Server
PRINTF ("Client are normal, it sends to server ", Number of Req.
/*****************************************************************/
```

PERSONAL PUBLICATIONS AND ARTICLES

Farea, A. H., & Küçük, K. (2022). Machine Learning-Based Intrusion Detection Technique for IoT: Simulation with Cooja *I. J. Computer Network and Information Security, (Ready for Submission)*

Farea, A. H., & Küçük, K. (2022). Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja. *EAI Endorsed Transactions on Internet of Things*, *7*(28), 1–12. https://doi.org/10.4108/eetiot.v7i28.324.

Farea, A. H., & Küçük, K. (2021). Multidisciplinary Studies and Innovative Technologies Enhancement Trust Management in IoT to Detect ON-OFF Attacks with Cooja. *Dergipark, 5*, 123–128.

Farea, A. H., & Küçük, K. (2021). Enhancement Trust Management in IoT to Detect ON-OFF Attacks with Cooja. *5th International Symposium on Multidisciplinary Studies and Innovative Technologies*, Turkey.

BIOGRAPHY

In 2012 he finished high school in Yemen. In 2019, he graduated from Taibah University in Madinah with a bachelor's degree in Computer Engineering with a GPA of 4.73 out of 5. During his studies, he worked as an IoT Full Stack Developer on integrated systems that serve pilgrims, passports, and military systems as well as drone systems. He is interested in the country and culture of Turkey. He wants to acquire Turkish expertise in many fields, especially in the techniques applied to daily systems. In 2019, he accepted and continued his studies at Kocaeli University in Turkey for a master's degree in computer engineering. His GPA is 3.73 out of 4. He is very interested in the Internet of Things and Artificial Intelligence in terms of research scope and daily applications that employ them. In addition to his studies, he developed his own web application for shopping for electronic devices.