Journal of the Faculty of Engineering and Architecture of Gazi University 40:3 (2025) 2087-2101

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Data-plane-based system and real-time attack prevention for software-defined networks distributed denial-of-service attacks

Kenan Erdoğan[1,2]*, Yasin Genç[2], NurselAkçam[2], Erkan Afacan[2]

[1]Graduate School of Natural and Applied Sciences, Gazi University, 06500, Ankara, Türkiye
[2]Department of Electrical-Electronic Engineering, Faculty of Engineering, Gazi University, 06570, Ankara, Türkiye

Highlights:

- Design of a system for the effective detection of DDoS attacks
- The results show that the MiddleModule system works properly under different test scenarios
- Mininet offers more detailed performance analysis results in a more realistic test environment

Keywords:

- Distributed Denial of Service Attacks
- 5G
- Software Based Networks
- Defense Systems
- MiddleModule Systems

Article Info:

Research Article
Received: 29.07.2024
Accepted: 06.04.2025

DOI:

10.17341/gazimmfd.1524120

Correspondence:

Author: Nursel Akçam
e-mail: ynursel@gazi.edu.tr
phone: +90 312 582 3343

Graphical/Tabular Abstract

This paper aims to clarify the fundamental problems of software-based networks (SDN) that can fail under Distributed Denial of Service (DDoS) attacks. A defense mechanism is offered that can effectively detect DDoS attacks. In the SDN's layer, it is aimed to create a real-time intrusion detection and prevention system in order to eliminate the deficiencies of the current security approaches for Internet of Things (IoT) networks. In this paper, system performance degradation test and defense strength test situations are examined in terms of OMNET ++ and Mininet. The test scenario shown in Figure A was created on the OMNET++ 4.2 platform.

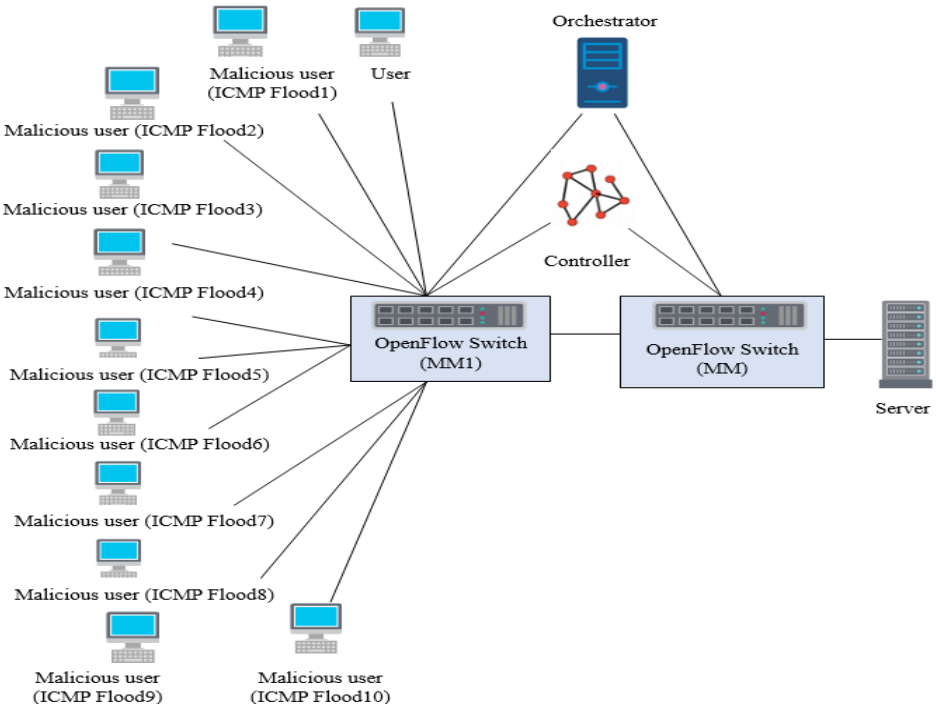


Figure A. System Performance Degradation Test Scenario within the Scope of OMNET++

Purpose: On the OMNET++ 4.2 simulation platform, defense force tests were applied to the MiddleModule system to achieve three objectives. In the defense strength test, basic functional analysis tests were applied to observe whether the MiddleModule system detects malicious activities.

Theory and Methods: Within the scope of the system performance degradation test, the impact of the normal user on the average packet receiving time for the MiddleModule system was measured. The testing was implemented using four different scenarios, with or without a MiddleModule system, and with or without malicious traffic generated by malicious user nodes.

Results: Within the scope of dataset modeling specific to SDNs, a realistic environment containing flow data for common network attacks as well as normal traffic is presented, and the results about the accuracy and performance of intrusion detection in the presence and absence of the proposed security are evaluated.

Conclusion: Both OMNET++ and Mininet-based tests gave similar results, especially in the functional analysis phase.

Journal of the Faculty of Engineering and Architecture of Gazi University 40:3 (2025) 2087-2101

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Yazılım tanımlı ağlar açısından dağıtılmış hizmet reddi doğrultusundaki saldırılara yönelik veri katmanı tabanlı sistem ve gerçek zamanlı saldırıların önlenmesi

Kenan Erdoğan[1,2], Yasin Genç[2], NurselAkçam[2]*, Erkan Afacan[2]
[1]Gazi Üniversitesi, Fen Bilimleri Enstitüsü, 06500, Ankara, Türkiye
[2]Gazi Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 06570, Ankara, Türkiye

ÖNEÇIKANLAR

- DDoS saldırılarını etkin olarak tespit eden sistem tasarımı
- Sonuçlar MiddleModule sisteminin farklı test senaryoları altında düzgün çalıştığını göstermektedir
- Mininet, daha gerçekçi bir test ortamında daha detaylı performans analiz sonuçları sunar

Makale Bilgileri
Araştırma Makalesi
Geliş: 29.07.2024
Kabul: 06.04.2025

DOI:
10.17341/gazimmfd.1524120

Anahtar Kelimeler:
Dağıtılmış hizmet reddi saldırıları, MiddleModule sistemi, yazılım tanımlı ağlar, siber savunma sistemleri, simülasyon ve test

ÖZ

Bu çalışmada, Dağıtılmış Hizmet Reddi (Distributed Denial of Service-DDoS) saldırıları karşısında olumsuz etkilenen Yazılım Tanımlı Ağ (Software Defined Network-SDN)'ların güvenliğine ilişkin DDoS saldırılarını etkin bir şekilde tespit edebilen savunma mekanizması önerilmektedir. SDN katmanında Nesnelerin İnterneti (Internet of Things-IoT) ağlarına yönelik hali hazırdaki güvenlik yaklaşımlarının eksikliklerini gidermek amacıyla, gerçek zamanlı bir saldırı tespit ve önleme sisteminin oluşturulması amaçlanmıştır. Hedeflenen gereksinimler göz önüne alınarak, bir SDN'ın her kenar anahtarının istatistiklerini topladığı ve gelen paketlere istatistiksel tabanlı algılama algoritmaları uyguladığı "MiddleModule" olarak adlandırılan bir sistem önerilmiştir. IP kandırma, protokol istismarı ve flood saldırıları gibi farklı DDoS türlerini tespit etmek üzere optimize edilmiş hafif algılama/tespit algoritmaları bu sistem içerisinde tasarlanmıştır. OMNET++ ve Mininet platformlarında gerçekleştirilen simülasyonlarda önerilen MiddleModule sisteminin savunma gücü performansı kapsamında duyarlılığı, yanlış pozitif oranı ve doğruluğu ölçülmektedir. Önerilen sistemin duyarlılık sonuçları tüm DDoS saldırı türleri için %99,6 ve doğruluk sonuçları %99,8'in üzerinde olduğu, paket alma süresindeki ek yükünün %0,2 olduğu ve yanlış pozitif oranının %0 olduğu tespit edilmiştir. Bu sonuçlar hedeflenen tüm Ağ/Aktarım katmanında DDoS saldırı türlerinin MiddleModule sistemi tarafından doğru bir şekilde algılandığı anlamına gelmektedir. Ayrıca önerilen MiddleModule sistemi literatürdeki mevcut veri katmanı tabanlı DDoS savunma sistemleri ile karşılaştırılmıştır.

Data-plane-based system and real-time attack prevention for software-defined networks distributed denial-of-service attacks

HIGHLIGHTS

- Design of a system for the effective detection of DDoS attacks
- The results show that the MiddleModule system works properly under different test scenarios
- Mininet offers more detailed performance analysis results in a more realistic test environment

Article Info
Research Article
Received: 29.07.2024
Accepted: 06.04.2025

DOI:
10.17341/gazimmfd.1524120

Keywords:
Distributed denial of service attacks, MiddleModule system, software defined networks, cyber defense systems, simulation and testing

ABSTRACT

In this study, a defence system that can effectively detect DDoS attacks is proposed for the security of Software Defined Networks (SDNs) that are adversely affected by Distributed Denial of Service (DDoS) attacks. In order to overcome the deficiencies of current security approaches for Internet of Things (IoT) networks at the SDN layer, it is aimed to create a real-time intrusion detection and prevention system. Considering the targeted requirements, a system called "MiddleModule" is proposed, where an SDN collects statistics of each edge switch and applies statistical-based detection algorithms to incoming packets. Lightweight detection algorithms optimized to detect different types of DDoS such as IP spoofing, protocol exploitation and flood attacks are designed within this system. In the simulations performed on OMNET++ and Mininet platforms, the sensitivity, false positive rate and accuracy of the proposed MiddleModule system are measured within the scope of the defence power performance. The sensitivity results of the proposed system are 99.6% for all DDoS attack types, the accuracy results are above 99.8%, the extra load on packet receive time is 0,2% and the false positive rate is 0%. These results indicate that all targeted Network/Transmission layer DDoS attack types are correctly detected by the MiddleModule system. Furthermore, the proposed MiddleModule system is compared with existing data layer based DDoS defence systems in the literature.

*Sorumlu Yazar/Yazarlar / *Corresponding Author/Authors: kenan.erdogan@csgb.gov.tr, yasin.genc@gazi.edu.tr, *ynursel@gazi.edu.tr, e.afacan@gazi.edu.tr / Tel: +90 312 582 3343

1. Giriş (Introduction)

Hizmet Reddi (Denial of Service-DoS) ve Dağıtılmış Hizmet Reddi (DDoS) saldırıları, ağ ve bilgi güvenliği alanlarındaki en önemli siber tehditlerin başında gelmektedir. DoS saldırıları, hatalı biçimlendirilmiş mesajlar göndererek saldırı yapılacak sunucunun/sistemin erişilebilirliğini bozmayı amaçlayan bir saldırı yöntemidir. Saldırganın bir sunucuya saldırması için finansal kazanç, intikam, ideolojik inanç, entelektüel meydan okuma veya siber savaş gibi nedenler etkili olabilir [1]. İyi niyetli olmayan saldırı birden fazla kötü niyetli kullanıcı tarafından desteklendiğinde, DoS saldırısı Dağıtılmış Hizmet Reddi yani DDoS saldırısına dönüşür. Bu kapsamda değerlendirildiğinde DDoS saldırılarının yapısı, DoS saldırılarına göre sistemlerde daha fazla zarar verebilecek anormal ağ trafiğinin ortaya çıkmasına neden olmaktadır. Bu tür bir saldırı altında, saldırıya uğrayan sunucu veya sistemler yasal kullanıcılar tarafından erişilemez hale gelir. Ayrıca ağ sistemi çok fazla paketle tıkanır ve ağ cihazları; arabellek taşmaları ve aşırı paket düşüşleri nedeniyle yasal kullanıcılara bazı hizmetleri ve hatta hiçbir hizmet veremez hale gelebilir [1, 2].

Yazılım tanımlı ağ (Software Defined Network-SDN), yeni bir ağ mimarisi ve paradigması olup 5G ve ötesi kapsamında haberleşme sistemlerinin gelişmesi ile birlikte birçok uygulamada kullanımının yaygınlaşması beklenmektedir. Bununla birlikte SDN'lerin merkezi kontrol mimarisi, bu ağların DDoS saldırılarına karşı savunmasız hale getirmektedir [3, 4]. DDoS saldırıları, ağ oluşturma alanında her zaman büyük bir problem olarak görülmektedir. Ancak SDN açısından DDoS saldırılarının etkileri geleneksel ağlardan daha yıkıcıdır. Bunun nedeni SDN'nin mimarisinden kaynaklanmaktadır. Bu mimaride, kontrol katmanı merkezi bir denetleyiciye dayanmaktadır. Denetleyici hedef alındığında, tüm ağ hizmetleri ciddi şekilde etkilenir veya tamamen durabilir. Geleneksel ağlarda bu risk daha azdır çünkü kontrol genellikle dağıtık bir mimari içermektedir. Ayrıca SDN, trafik akışlarını dinamik olarak yönlendirir. Ancak saldırı sırasında bu yönlendirme yeteneği işlevsiz hale geleceğinden ağın yönetimi çok ciddi zarar görecektir. Bu sorun, gelecekte devasa boyutta artması beklenen nesnelerin interneti (Internet of Things-IoT) ağına bağlı cihazlarının karmaşık trafik yapısı ve yüksek bant genişliği gereksinimleri ile daha kritik bir boyut kazanacaktır [5].

SDN kapsamında DDoS'ne yönelik savunma yöntemleri ilgi çekici bir çalışma alanı olmaya devam etmekte ve bu güvenlik problemine yönelik kabul edilen kesin bir çözüm yöntemi bulunmamaktadır [6]. Mevcut çalışmalar incelendiğinde [6-11], önerilen savunma sistemlerinin ya belirli saldırı türlerine odaklandığı ya da gerçek zamanlı performans sağlamada yetersiz kaldığı görülmektedir. Ayrıca, bazı savunma sistemlerinin saldırılara karşı düşük algılama oranları ve yüksek yanlış pozitif oranları gibi eksikleri olduğu görülmektedir. Örneğin, SYN Flood saldırılarına odaklanan algoritmalar, IoT cihazlarının neden olduğu daha karmaşık trafik modellerini ele almakta zorluk çekmektedir [12].

Gerçek zamanlı saldırıların önlenmesi doğrultusunda mobil ağlar, büyük veri ve bulut bilişim alanlarındaki gelişmeler sayesinde, nesnelerin internetine bağlı IoT cihazlarının sayısı her geçen gün artmaktadır [7]. Günlük yaşamı kritik derecede etkileyen ve kolaylaştıran IoT kullanım örnekleri arasında akıllı evler, otonom arabalar, güvenlik sistemleri, akıllı şehirler ve uzaktan sağlık hizmetleri bulunur [13]. IoT tarafından üretilen büyük hacimli veriler göz önünde bulundurulduğunda, bu çeşitli kullanım durumlarının hizmet kalitesi gereksinimlerinin geçmişten günümüze gelen kablosuz ağlar tarafından karşılanamayacağı açıktır. Kaynak yönetimi için SDN ve Ağ Fonksiyonlarını Sanallaştırmasına (Network Function Virtualization - NFV) dayanan 5G ve ötesi ağlar, gelecekte her yerde bulunacak olan IoT için önemlidir.

Etkili model oluşturmaya yönelik gerçekçi ağ trafiği verilerinin önemine karşın, IoT saldırı tespitindeki mevcut araştırmaların çoğu, IoT trafiği olmayan eski ağlar için oluşturulmuş veri kümeleri kullanır. Bu çoğunlukla yakın zamanda yayınlanan Bot-IoT veri seti haricinde IoT trafiğini içeren halka açık veri setlerinin yer almamasından kaynaklanmaktadır [8]. Özellikle SDN kapsamındaki IoT'ye dair ortamlar için halka açık herhangi bir veri kümesi bulunmaz. IoT ve SDN ağ trafiği nitelikleri, eski ağlara göre epeyce farklılık gösterirler. Bu nedenle eski ağ verileriyle desteklenen modellerin kullanılması yanlış sınıflandırma sonuçlarına sebebiyet verebilir.

Bu çalışmada; IoT ağları için mevcut güvenlik yaklaşımlarının eksikliklerini gidermek amacıyla, SDN veri katmanında gerçek zamanlı bir saldırı tespit ve önleme sisteminin tasarımı gerçekleştirilmiştir. Bu sistem, denetleyiciden geçen trafik akışlarını izleyen ve makine öğrenimine dayalı akıllı bir saldırı algılama modülü kullanarak trafiği sınıflandıran, uçtan uca bir saldırı tespit ve önleme sistemi mimarisini kurarak, SDN için yeni bir güvenlik yaklaşımı sunmayı amaçlamıştır. Bu kapsamda, SDN üzerinde DDoS saldırılarının etkileri ve mevcut savunma yöntemleri incelenerek mevcut sistemlerdeki eksiklikler belirlenmiştir. Bunun sonucu olarak önerilecek olan sistemin özellikleri ve gerekliliklerini belirlenerek SDN için veri katmanı tabanlı ve "MiddleModule" olarak adlandırılan bir savunma sisteminin tasarım gerçekleştirilmiştir. MiddleModule sistemi ile her SDN'ın uç anahtarları, istatistikler toplayarak ve istatistiksel tabanlı algılama algoritmaları uygulayarak kötü amaçlı etkinlikleri algılar. Bu savunma sistemi, çoklu konum savunma yaklaşımı kullanarak farklı konumlarda farklı algılama işlevleri gerçekleştirerek, etkili bir şekilde çalışır. Ayrıca, önerilen savunma sistemi kötü amaçlı trafikleri SDN denetleyicisine göndermemektedir, böylece aşırı yüklenme ve tıkanıklık sorunları önlenmektedir. Önerilen MiddleModule sistemi için kenar anahtarda kullanılmak üzere "Paket Aynası ve Saldırı Önleme Bloğu", "Paket İşleme Bloğu" ve "Kontrol Bloğu" tasarlanmıştır. Ayrıca önerilen sistemin farklı türdeki DDoS saldırılarına karşı koyabilmesi için saldırı tespit algoritmaları tasarlanmıştır.

Bununla birlikte, makine öğrenimi tabanlı bir saldırı tespit ve önleme sistemi mimarisi de sunulmuştur. Veri setlerinin seçilmesinde ve kullanılması hususu oldukça önemli olduğundan IoT ağları için yeni bir veri seti tasarlanmıştır. Bu veri seti, normal trafik ve farklı saldırı trafiği türlerini içermektedir ve bu sayede sistemin performansı gerçek dünya senaryolarına daha iyi uyum sağlayacak şekilde test edilmektedir.

Önerilen sistem literatürde mevcut Avant-Guard, CIDS, StateSec ve SDNScore veri katmanı tabanlı DDoS savunma sistemleri ile karşılaştırılmıştır. OMNET++ ve Mininet platformlarında gerçekleştirilen simülasyonlarda önerilen MiddleModule sisteminin savunma gücü performansı kapsamında duyarlılığı, yanlış pozitif oranı ve doğruluğu ölçülmektedir. Önerilen sistemin duyarlılık sonuçları tüm DDoS saldırı türleri içinde duyarlılık sonuçları %99,6, doğruluk sonuçları %99,8'in üzerinde olduğu ve yanış pozitif oranının %0 olduğu test sonuçlarında görülmektedir. Bu sonuçlar hedeflenen tüm Ağ/Aktarım katmanında DDoS saldırı türlerinin MiddleModule sistemi tarafından doğru bir şekilde algılandığı anlamına gelmektedir. Ayrıca MiddleModule sisteminin kullanılmasından kaynaklanacak paket alma süresindeki ek yükünün %0,2 olduğu hesaplanmıştır. Bu nedenle, paket alma süresi açısından MiddleModule sisteminin sistem performansı düşüşü ihmal edilebilir düzeydedir. Bu testlerde OMNET++ platformunda NSL-KDD veri seti ve Mininet platformunda ise MAWI Working Group Traffic Archive'den [14] alınan gerçek bir trafik veri seti kullanılmıştır.

Bu savunma sistemlerden Avant-Guard sistemi [9] sadece SYN Flood DDoS saldırı türünü dikkate alır, CIDS sistemi [10] SYN Flood saldırılarını ve Flooding saldırılarını (ICMP Flooding ve UDP Flooding) dikkate alır, StateSec sistemi [11] ise sadece Flooding saldırılarına karşı test edilir. SDNScore [6], SYN Flood saldırısı ve Flooding saldırılarına odaklanmaktadır. Literatürdeki mevcut sistemlere kıyasla önerilen MiddleModule sistemi farklı DDoS saldırılarına karşı daha kapsamlı bir savunma ve daha etkin performans sunmaktadır.

Makalenin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2, literatürdeki ilgili çalışmaları sunar. Bölüm 3'te yazılım tanımlı ağ (SDN) mimari incelenmiştir. Bölüm 4'te önerilen MiddleModule sisteminin mimarisi ve tasarlanan alt bloklar konusu işlenmiştir. Bölüm 5'te MiddleModule sistemi için tasarlanan tespit algoritmaları tanıtılarak tespit algoritmalarının ayrıntıları ve işleyişleri sunulmuştur. Bölüm 6'da MiddleModule sisteminin performans analizleri gerçekleştirilmiştir. Son bölümde ise çalışma ve testlerden elde edilen sonuçlar verilmiştir.

2. İlgili Çalışmalar (Related Works)

Ağlarda saldırı tespiti ve etkilerinin azaltılması, IoT tarafından oluşturulan büyük saldırı yüzeylerinin neden olduğu artan siber güvenlik olayları ile her zamankinden daha önemli bir araştırma konusu haline gelmiştir. Son on yılda SDN temelli ağ mimarilerinin giderek daha fazla benimsenmesiyle birlikte SDN güvenliği, siber güvenlik araştırma topluluğu için ilgi merkezlerinden biri haline gelmiştir. Bu kapsamda literatür detaylı olarak taranarak SDN'nin güvenliğini sağlamak için gerçekleştirilen çalışmalar incelenmiştir.
Rathore ve Park [15], IoT ağlarında dağıtılmış saldırı tespiti için sis tabanlı yarı denetimli bir öğrenme yaklaşımı önermektedir. Araştırmacılar, NSL-KDD veri setini kullanmış ve dağıtık yaklaşımlarının algılama süresi ve doğruluğu açısından merkezi çözümden daha iyi performansa sahip olduğunu göstermişlerdir. Evmorfos, Vlachodimitropoulos, Bakalos ve Gelenbe [16] tarafından, IoT ağlarında SYN taşma saldırılarını algılamak için Rastgele Sinir Ağı ve Uzun Kısa Süreli Bellek (Long Short Term MemoryLSTM) kullanan bir model önerilmiştir. Soe, Feng, Santosa, Hartanto ve Sakurai [17] tarafından, IoT ağları için üç makine öğrenimi modeli kullanan sıralı bir saldırı tespit yapısı önerilmektedir. Araştırmacılar, N-BaIoT veri setini kullanmakta ve %99 doğruluk elde etmektedirler. Alqahtani, Mathkour ve İsmail [18] tarafından, nesnelerin interneti ağlarındaki özellikleri seçmek için Fisher skorunu kullanan, genetik tabanlı bir aşırı gradyan artırma (GXGBoost) modeli önerilmiştir. Araştırmacılar ayrıca N-BaIoT veri setini kullanmışlar ve %99,96 doğruluk elde etmişlerdir. Bu yaklaşımlar IoT'ye yönelik saldırıları başarılı bir şekilde tespit etse de geleneksel ağlarda çalışan uygulamalar olarak görülmektedirler.

Şimdiye kadar SDN ağlar için önerilen çözümlerin çoğu, DoS ve DDoS saldırılarının tespit edilmesi ve hafifletilmesine yönelik tekniklere odaklanmaktadırlar. Bunun nedeni SDN mimarisin DDoS saldırıları karşısında olumsuz etkilenmesi ve bu sorununun çözümünün önem arz etmesidir. Ravi ve Shalinie [19] SDN tabanlı IoT ağlarındaki DDoS saldırılarını tespit etmek için UNB-ISCX veri kümesi üzerinde yarı denetimli bir model kullanmıştır. Model, UNB-ISCX veri kümesinde ve yalnızca UDP flood saldırılarını içeren araştırmacıların kendi veri kümesinde %96'nın üzerinde doğruluk elde etmişlerdir. Galeano-Brajones, Carmona-Murillo, Valenzuela-Valdés ve LunaValero [20] tarafından, yazılım temelli IoT ağlarında DoS ve DDoS saldırılarının tespiti ve hafifletilmesi için entropi tabanlı bir çözüm önerilmektedir. Yaklaşım, Bot-IoT veri kümesinde ve araştırmacıların TCP SYN sel saldırılarını içeren veri kümesinde yüksek doğruluk elde etmişlerdir. Amangele, Reed, Al-Naday, Thomos ve Nowak [21], SDN kullanan IoT ağlarındaki anormallikleri tespit etmek için CART Karar Ağacı (KA) algoritması önermektedir. Araştırmacılar CICIDS'2017 veri setini kullanmış ve %99 tespit oranı elde etmişlerdir. Derhab ve arkadaşları [22], endüstriyel kontrol süreçlerini hedef alan saldırılara karşı Rastgele Alt Uzay Öğrenme, k En Yakın Komşu (kEYK) ve blockchain altyapısı kullanan bir izinsiz giriş tespit sistemi önerilmiştir. Araştırmacılar, endüstriyel kontrol sistemi siber saldırı veri setini kullanmakta ve çözümlerinin yüksek doğruluk sağladığını ortaya koymaktadırlar. Açıklanabilir saldırı tespit sistemlerindeki çalışmalar şu ana kadar oldukça sınırlı kalmıştır.

Wang, Zheng, Yang ve Wang [23] tarafından, Shapley Ek Açıklamalarına dayalı saldırı tespit sistemleri için açıklanabilir bir makine öğrenimi çerçevesi önerilmiştir. Bu çalışma, NSL-KDD veri seti üzerinde test edilmiştir. Gündüz ve Sağıroğlu [24] tarafından, çalışmalarında kullanıcıların farklı güvenlik hizmeti aboneliklerine sahip olduğu bir 5G ağındaki trafiğin, enerji etkin bir biçimde yerleştirilmiş olan sanal ağ güvenliği fonksiyonları (Virtual Network Security Functions -VNSF) üzerinden güvenlik ihtiyaçları doğrultusunda yönlendirilmesi amaçlanmıştır. Bu amaçla, 5G altyapısında enerji tüketimini en aza indirgemeyi hedefleyen bir sezgisel algoritma önerilmiş ve algoritmanın aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendiren yeni bir SDN kontrolcüsü tasarlanmış ve geliştirilmiştir. Geliştirilen SDN kontrolcüsünün ağa gelen trafiği toplam bant genişliği, uçtan uca gecikme ve paket kayıp oranı gibi hizmet kalitesi parametreleri açısından etkin bir şekilde yönlendirebildiği görülmüş ve ulaşılan maksimum kayıp oranı değeri %0,4 olarak ölçülmüştür. Tonkal ve Polat [25] tarafından, SDN'da makine öğrenme modellerini kullanarak bir trafik sınıflandırma yaklaşımı önerilmiştir. Dağıtık İnternet Trafik Oluşturucu (D-ITG) aracı kullanılarak SDN üzerinde DNS, Telnet, ping ve ses trafik akışları oluşturulmuş ve bu trafik akışlarını temsil eden on iki öznitelik (iletilen paket sayısı, ortalama iletim süresi, anlık iletilen paket sayısı vb.) belirlenmiş ve fiziksel ağdaki SDN kontrolcüsü üzerinden gerçek zamanlı olarak özniteliklere ait veriler toplanarak bir veri seti oluşturulmuştur. Daha sonra da bu veri seti üzerinde trafik sınıflandırması için kEYK, Destek Vektör Makinesi (DVM), Çok Katmanlı Algılayıcı (ÇKA), KA ve Naive Bayes (NB) makine öğrenme modellerinin başarımı test edilmiştir. Gerçek zamanlı olarak oluşturulan bu veri seti üzerinde kEYK modeli kullanıldığında %99,4 doğruluk oranı ile en yüksek sınıflandırma doğruluğu elde edilmiştir.

3. 5G Ağları Kapsamında Yazılım Tanımlı Ağlar (Software-Defined Networks within the Scope of 5G Networks)

SDN uygulama katmanı, kontrol katmanı ve veri katmanı gibi farklı katmanlara ayrılarak geleneksel ağdan farklı bir ağ mimarisi önermesiyle yeni bir uygulama alanı olarak değerlendirilir [3, 26, 27]. Geleneksel ağlarda, yönlendirme ve iletme olmak üzere iki temel ağ oluşturma süreci bulunur. Yönlendiriciler ağ boyunca dağıtılırlar ve birbirleriyle iletişim kurarak bağımsız bir şekilde çalışırlar. SDN dahilinde bu iki işlem ise farklı ağ katmanlarında dağıtılan farklı cihazlarda işlenir. SDN, veri ve kontrol katmanlarını birbirinden ayırdığından diğer ağ biçimlerinden farklı olarak görülür; burada yönlendiriciler/anahtarlar, artık iletme işlevinden sorumludur ve yönlendirme kararları denetleyici (kontrol katmanı) tarafından alınır [28].

SDN mimarisinde yük dengeleme ve izinsiz giriş tespit sistemleri gibi tüm uygulamalar, uygulama katmanında çalışır ve denetleyici ile iletişim, kuzeye bağlı Uygulama Programlama Arayüzü (Application Programming Interface-API) üzerinden gerçekleşir. Denetleyici ve anahtarlar arasındaki iletişim, çoğunlukla OpenFlow protokolü [29] kullanılarak güneye bağlı API aracılığıyla gerçekleşir. Mantıksal olarak merkezileştirilmiş denetleyici, ağı yönetmekten sorumludur. Denetleyici, ağın genel bir görünümünü korur ve verdiği

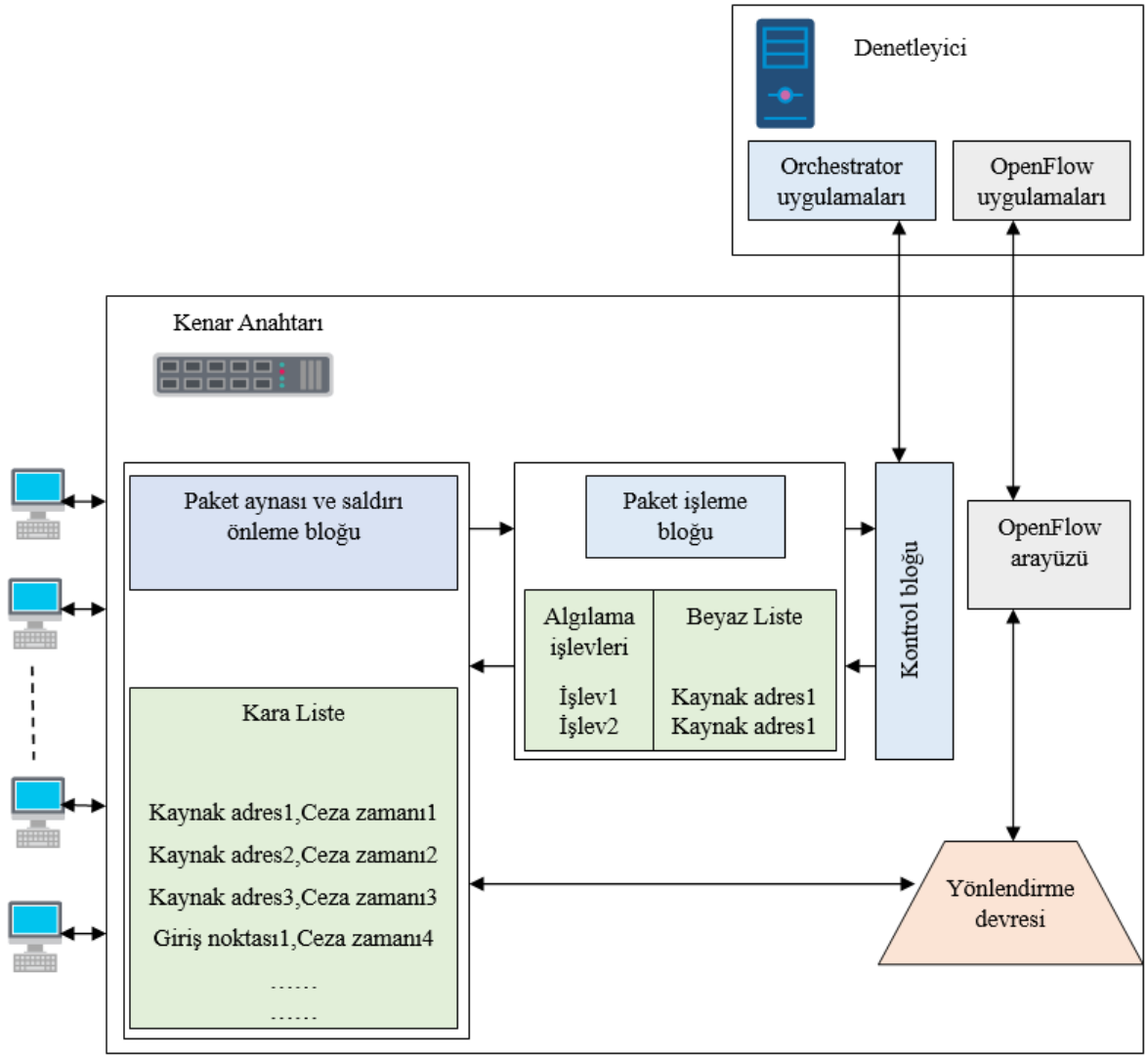
yönlendirme kararlarına dayalı olarak ilgili anahtarlara akış kuralları adı verilen yönlendirme kuralları yükler. Anahtarlar, akış kurallarını akış tablolarında depolar ve mevcut akış kurallarıyla eşleşmelere dayalı olarak ağ paketlerini iletirler [30]. SDN ağları uzaktan ve dinamik olarak yönetmek için gereken ağ sanallaştırma yeteneklerini sağlayarak 5G ve ötesi ağların en büyük etkilerinden biri olacağı düşünülmektedir. SDN uygulamalarıyla elde edilecek hızlı yük devretme ve otonom yönetim yetenekleri, 5G ağlarının yüksek bant genişliği ve düşük gecikme gereksinimlerini sağlayarak, çeşitli IoT kullanım durumlarını desteklemelerini sağlar [31]. SDN'ların, NFV ile birlikte, özellikle 5G çekirdek ve radyo erişim ağlarında ağ dilimlemenin temelini oluşturacağı ve böylece operatörlerin gerekli hizmet kalitesini sağlamak için altyapılarını verimli bir şekilde kullanmalarına önemli bir olanak sağlayacağı ve müşterilerine güvenlik garantisi sunmak için gerekli işlevleri yerine getireceği düşünülmektedir [32].

4. Önerilen MiddleModule Sisteminin Mimarisi (Architecture of the Proposed MiddleModule System)

Bu çalışma kapsamında önerdiğimiz MiddleModule sistemi; SDN'nin siber güvenliğini sağlanmasında izleme, tespit ve önleme amaçlarını veri katmanında; orkestrasyon ve konfigürasyon amaçları içinde kontrol katmanında her biri farkı göreve sahip blok ve algoritmalardan oluşmaktadır. Önermiş olduğumuz MiddleModule sisteminde kenar anahtarlarda kullanılan sistem blokları yalnızca basit ama etkili karar verme süreçlerini işletmektedir. Bu amaçla kenar anahtarda kullanılmak üzere "Paket Aynası ve Saldırı Önleme Bloğu", "Paket İşleme Bloğu" ve "Kontrol Bloğu" tasarlanmıştır. Bu bloklar; saldırı tespiti ve önleme işlemlerinin gerçek zamanlı olarak gerçekleştirilmesi için tasarlanmıştır. Bu blokların gerçekleştirdiği görevler sayesinde denetleyici üzerindeki yük azaltılmakta ve SDN'nin genel performansı iyileştirilmektedir. MiddleModule sisteminin tasarımında kullanılan bloklar Şekil 1'de gösterilmiş ve her bir bloğun görevi aşağıda incelenmiştir.

4.1. Veri Katmanında MiddleModule Blokları (*MiddleModule Blocks in Data Layer)*

Önerilen MiddleModule sistemi; izleme, algılama ve hafifletme yetenekleri sağlamak için uç anahtarlara "Paket Aynası ve Saldırı Önleme Bloğu" ve "Paket İşleme Bloğu" gibi birkaç blok eklenmiştir. Kenar anahtarların tespit ve önleme etkinliklerinde denetleyici ile iş birliği için kenar anahtarlara "Kontrol Bloğu" da eklenir. Bundan dolayı her kenar anahtarına toplamda üç işlevsel blok eklenmiştir. Şekil 1'de bu blokların yerleşimi ve bağlantıları gösterilmektedir.



Şekil 1. MiddleModule blokları (MiddleModule blocks)

4.1.1. Paket Aynası ve Saldırı Önleme Bloğu
(Packet Mirror and Anti-Intrusion Block)

"Paket Aynası ve Saldırıyı Önleme Bloğu", gelen paketlerin "Paket İşleme Bloğu" kısmına yansıtılmasında kaynak adresi ya da giriş bağlantı noktası kötü amaçlı listede bulunuyorsa gelen paketin bırakılması amacıyla iki temel işlem gerçekleştirir. Bir paket bir kenar anahtarına girdiğinde doğrudan bu bloğa gider ve ilk olarak "Saldırı Önleme Alt Bloğu" paketini gözden geçirir. Bu paketin kaynak adresi kötü amaçlı listede bulunuyorsa, bu paket başka bir bloğa gönderilmeden bırakılır. Listede yoksa "Paket Aynası Alt Bloğu" paketini alır. Bu alt blokta, orijinal paketin tamamı iletme devresine yönlendirilirken, paketin başlık kısmı "Paket İşleme Bloğu" aracılığıyla yansıtılır.

"Saldırı Önleme Alt Bloğu", kötü amaçlı yazılım tespit sonuçlarına göre uygun azaltma eyleminin gerçekleştirilmesinden sorumludur. Şekil 1'de görüldüğü gibi, MiddleModule sistemine yönelik olarak gelen paketi alan ilk blok "Saldırı Önleme Alt Bloğu" dur. Bu durumun sebebi, alınan paket zaten kayıtlı bir kötü niyetli kullanıcıdan geliyorsa, bu paketi tekrardan analiz edilmesine gerek duyulmamasıdır. Bu alt blok, önceden kaydedilmiş kötü amaçlı kullanıcıların kaynak adreslerini ya da giriş bağlantı noktalarını içeren bir kötü amaçlı listeye sahiptir.

4.1.2. Paket İşleme Bloğu (Package Processing Block)

"Paket İşleme Bloğu", kenar anahtarlara algılama yeteneği sağlar. Bundan dolayı tespit işlemlerine yönelik olarak kenar anahtarlara eklenen nitelik şeklinde ifade edilir. Bu blok, "Paket Yansıtma Alt Bloğu" kısmından paket başlıklarını alır ve birkaç DDoS algılama algoritması kullanılarak bunlar işlenir. Bu nedenle, "Paket İşleme Bloğu", Giriş Kuyruğu ve Algılama Motoru gibi işleme ve depolama altyapısını içeren birkaç alt modül içerir. Herhangi bir paketin başlık kısmı, "Paket İşleme Bloğu" tarafından alındığı zaman Giriş Kuyruğu içerisinde depolanır. Kullanım kolaylığı nedeniyle FIFO yani "ilk giren ilk çıkar" kuyruğu olarak tasarlanması tercih edilir. "Paket İşleme Bloğu" içerisinde akış tabanlı özellikler kaydedilir ve bu kayıtlar kötü amaçlı tespit algoritmalarında kullanılır. Akışlar, tespit algoritmasına bağlı olarak kaynak veya hedef adreslerine göre oluşturulur. Farklı algılama algoritmaları için farklı özellikler kaydedilir.

4.1.3. Kontrol Bloğu (Control Block)

"Kontrol Bloğu", kenar anahtarlarda çalışan MiddleModule blokları ile denetleyici üzerinde çalışan MiddleModule blokları arasındaki iletişim için kullanılır. Bu blok, MiddleModule sistemi için önem taşır. Çünkü sistem SDN'lar çerçevesinde tasarlanır. Tablo 1'de MiddleModule Mesaj Protokolünün bilgileri verilmiştir. Denetleyici, yönlendirme ile ilgili tüm kararlara karar verir. Bir kenar anahtarı herhangi bir kötü amaçlı etkinlik algılayıp paketleri bırakmaya başladığında, denetleyicinin bu etkinliğin farkında olması gerekir. Denetleyici herhangi bir kenar anahtarına bir beyaz liste veya kara liste tanımlayabilmelidir. Bunun yanı sıra denetleyici, ağdaki herhangi bir kenar anahtarının tüm MiddleModule işlemini etkinleştirebilmeli, devre dışı bırakabilmeli veya algılama algoritması değişkenlerini yapılandırabilmelidir.

4.2. Kontrol Katmanında MiddleModule Blokları
(MiddleModule Blocks in the Control Layer)

4.2.1. Orchestrator Bloğu (Orchestrator Block)

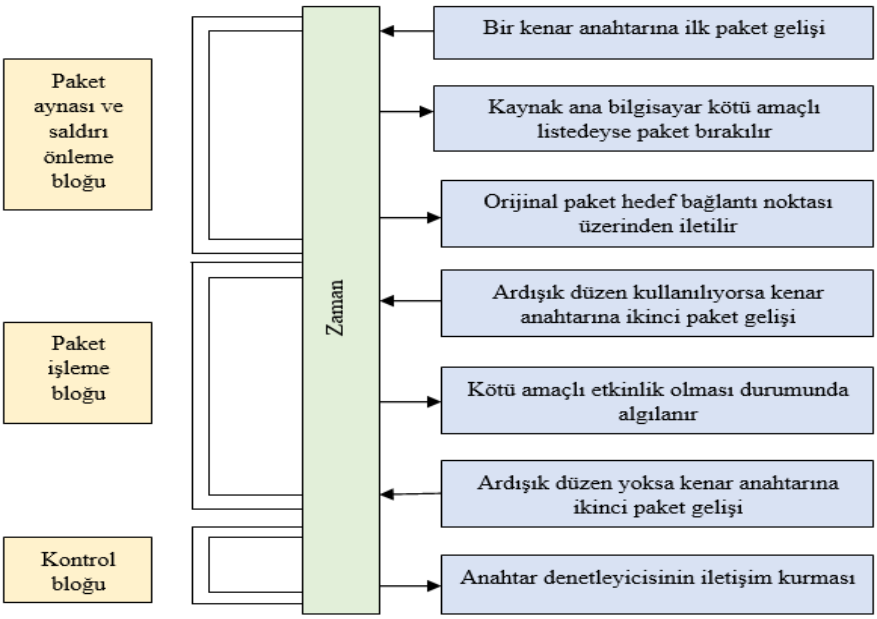
Orchestrator Bloğu, uç anahtarların Kontrol Blokları ile iletişim kurması için kullanılır. Bu bloğun iletişim davranışı, Kontrol Bloğu bölümünde açıklanan durumlara benzerlik gösterir. Orchestrator Bloğu, denetleyicinin üzerinde çalışan bir uygulama olarak tasarlanır. Bu blok, ağ operatörüne uç anahtarlarda çalışan MiddleModule savunma sistemini yapılandırma veya kontrol etme yeteneği sağlar. Bunun yanı sıra şebeke operatörü, MiddleModule sistemi tarafından verilen algılama kararlarını gözlemleyebilir. Ağ operatörü, sistemin performansını ayarlamak veya sistemin davranışını değiştirmek için algılama algoritmalarına eşik değerlerini yeniden atayabilir, ceza süresini değiştirebilir ve boşta kalma süresi ekleyebilir. Bu amaçla kullanılan eşik değerleri hem anormal hem de normal trafik özellikleri dikkate alınarak seçilir. Aslında, yanlış pozitifler açısından güvenli tarafta olmak için uygulanan eşik değerler, hesaplanan eşik değerler üzerinde bir değer dikkate alınarak seçilmektedir. Bu değerler, çalışma sırasında önerilen sistemin en düşük yanlış pozitif oranı ve en yüksek doğruluk oranını sağladığı noktalarda optimize edilmiştir. Eşik değerin seçimiyle ilgili [33] incelenebilir. Denetleyici bir şekilde aşırı yüklenirse, Orchestrator Bloğu sonlandırılabilir. Bu durum MiddleModule sisteminin genel performansını önemli ölçüde etkiler.

4.3. MiddleModule İşlemleri (MiddleModule Operations)

Bir paket bir kenar anahtarına ulaştığında, Şekil 2'de gösterildiği gibi doğrudan Paket Aynası ve Saldırı Önleme Bloğu kısmına aktarılır. Saldırı Önleme Alt Bloğu, paketi alır ve kaynak adresin kötü amaçlı listede olup olmadığını kontrol eder. Listede varsa, paket daha fazla analiz yapılmadan bırakılır. Bu, anahtar kaynaklarını, denetleyici kaynaklarını ve kurban ana bilgisayarı kötü amaçlı etkinliklerden korur. Ayrıca, bu paketin daha ayrıntılı analizi için hiçbir MiddleModule kötü amaçlı tespit kaynağı kullanılmaz. Paketin kaynak ana bilgisayarı kötü amaçlı listede değilse, Paket Aynası Alt Bloğu paketi alır ve Paket İşleme Bloğu aracılığıyla paketin başlık kısmını yansıtır ve paketin tamamını Yönlendirme Devresi aracılığıyla gönderir.

Tablo 1. MiddleModule mesaj protokolü (MiddleModule message protocol)

İsim	Mesaj ID	Gönderici ID	Alıcı ID	Kötü amaçlı kaynak adresleri
Boyut	2 Bayt	4 Bayt	4 Bayt	50 Bayt
Numara Bilgisi	0x01: İşlevlerin devre dışı bırakılması 0x02: Kötü amaçlı kaynak adresleri hakkında denetleyiciye bilgi verilmesi 0x03:Kötü amaçlı kaynak adreslerinin dağıtılması 0x04: Bağlantı noktasının algılanması	Her cihazın benzersiz numarası	Her cihazın benzersiz numarası	



Şekil 2. MiddleModule işlemleri diyagramı (MiddleModule operations diagram)

Tablo 2. IP kandırma tespit algoritması (IP spoofing detection algorithm)

Satır No	Algoritma
1	Yeni paket (p) girişi
2	s <= Paketin IP adresi kaynağı
3	i <= Paketin giriş anahtarı bağlantı noktası
4	Eğer i bir ana bilgisayar bağlantı noktası ise,
5	$pcktWindow_i[lastEntry_i] <= p$
6	Eğer $lastEntry_s < windowSize$ ise,
7	$lastEntry_i$ artar.
8	veya
9	$lastEntry_i <= 0$
10	$t_i <= pcktWindow_i$ listesindeki toplam paket sayısı
11	$diffSrc_i <= pcktWindow_s$ listesindeki toplam kaynak IP adresi paket sayısı
12	Eğer ($diffSrc_i > spoofThreshold$) ve ($t_i$ / $windowSize > saturationThreshold$)
13	p paketi kötü amaçlıdır
14	$t_i <= 0$
15	$açık\ (diffSrc_i)$

Sistemde organizasyon kullanılıyorsa, Paket Aynası ve Saldırı Önleme Bloğu boşta kaldığından, sistem ağdan yeni bir paket alabilir ve işleyebilir. Yönlendirme Devresi, anahtarın OpenFlow ve Yönlendirme bileşenlerinde MiddleModule sistemi herhangi bir değişiklik yapmadığı için standart bir OpenFlow anahtarında olduğu gibi çalışır.

Paketin başlık kısmı Paket İşleme Bloğuna ulaştığında, Giriş Kuyruğunda depolanır. Paket İşleme Bloğundaki Tespit Motoru her boşta kaldığında, Giriş Kuyruğundaki ilk giriş kuyruğundan çıkarılır ve Tespit Motoru girişi, başlık kısmını alır. Önce, Algılama Motoru giriş için karşılık gelen akışı bulur, ardından Özellikler Tablosunu kontrol eder ve alınan başlığın ilgili niteliklerini alır. Bu özellikler, Tespit Algoritmalarında kullanılan akış için alınan paketlerin önceki geçmişine dayanan listeler ve değerler içerir. Kaydedilen özelliklerle ilgili ayrıntılı açıklamalar, makalenin Algılama Algoritmaları bölümünde bulunabilir. Algılama Motoru, birkaç hafif Algılama Algoritması çalıştırır ve bunlardan herhangi biri kötü amaçlı etkinlik tespit ederse, paketin kaynak adresini kara listeye kaydeder. Kaynak ana bilgisayarın adresi zaten kara liste tablosundaysa, Tespit Motoru bu adresi Saldırı Önleme alt bloğuna göndererek başka bir ceza süresi için Kötü Amaçlı Listede saklamasını sağlar ve bu adresi kara liste tablosundan kaldırır. Algılama Motorundaki Özellikler Tablosu en son özelliklerle güncellenir. Kötü amaçlı bir etkinlik tespit edilirse, tespit kararı Kontrol Bloğuna da iletilir. Bu blok daha sonra bu kötü amaçlı etkinliğin kaynak adresini denetleyiciye gönderebilir ve denetleyici bunu tüm uç anahtarlara dağıtabilir.

5. Tespit Algoritmaları (Detection Algorithms)

MiddleModule sisteminde, ağ/aktarım katmanında DDoS saldırılarını tespit etmek için, uç anahtarların Paket İşleme Blokları içinde çeşitli tespit algoritmaları uygulanmaktadır. Başlıca beş tür ağ/aktarım katmanında DDoS Saldırısı vardır ve iyi bilinen bu saldırılara karşı aşağıdaki algılama algoritmaları tasarlanmıştır. Bu tespit algoritmaları aşağıda sunulmuştur.

5.1. IP Kandırma Saldırısı Tespit Algoritmaları
(IP Spoofing Attack Detection Algorithms)

IP kandırma saldırısı; kötü amaçlı bir ana bilgisayar, sahte kaynak IP adreslerine sahip paketler oluşturması temeline dayanmaktadır. Bu tür bir saldırıyı tespit etmek için, gelen trafiğin kaynak IP adresleri izlenerek gelen paketlerin kaynak IP adreslerine yönelik

varyasyondaki herhangi bir önemli artış tespit edilir. Tek bir ana bilgisayar bağlantı noktasının girişinden gelen farklı kaynak IP adreslerine sahip paket sayısı bir eşik değerinin üzerine çıkarsa, bu ana bilgisayar girişine bağlı IP Kandırma saldırganlarını kara listeye ekler. Tablo 2'de gösterildiği gibi IP kandırma saldırılarına karşı bir tespit/algılama algoritması tasarlanmıştır.

5.2. Protokol İstismar Saldırısı Tespit Algoritmaları (Protocol Exploit Attack Detection Algorithms)

Protokol istismar saldırısı kapsamında kötü amaçlı bir ana bilgisayar, kurban ana bilgisayarın kaynaklarını tüketmek için iletişim protokollerinin bazı özelliklerini kötüye kullanan paketler üretir. Bu açıdan, SYN Flood saldırısı, protokol istismar saldırı türünün bir örneği olarak ele alınarak Tablo 3'te açıklandığı gibi, SYN Flood saldırılarına karşı bir tespit algoritması geliştirilmiştir. SYN Flood saldırısını tespit etmek için, paketlerin kaynak tarafında farklı kaynak IP adreslerine sahip her bir ana bilgisayar için SYN paket sayısının toplam paket sayısına oranı hesaplanır. Dolayısıyla bu algoritma, mesaj tipindeki değişimi analiz eder.

Normal TCP (Transmission Control Protocol) trafiği için, SYN mesajlarının sayısı ile toplam mesaj sayısının oranı bir eşikten küçük olmalıdır. Eşik, ağ anormalliklerini kapsayacak şekilde dikkatlice seçilirse, bu oran eşiği aştığında, bir ana bilgisayardan bir kurban sunucu aracılığıyla bir SYN Flooding etkinliği olduğu sonucuna varılabilir. SYN Flood algılama algoritmasında, bir kenar anahtarına bir paket (p) ulaştığında, paketin bir ana bilgisayar bağlantı noktasından mı yoksa bir ağ bağlantı noktasından mı geldiğini görmek için paketin gelen bağlantı noktası (i) kontrol edilir. Bir ana bilgisayar bağlantı noktasından geliyorsa, p paketi, $lastEntry_s$ dizin numarasıyla $pcktWindow_s$ adlı bir listede depolanır. Farklı kaynak IP adresleri için farklı listeler kullanılır. Liste doluysa, yeni paketin üzerine en eski paket yazılır. Bu nedenle, liste her zaman algılama performansı için kritik olan en son paketleri içerir. Paket saklandıktan sonra, $pcktWindow_s$ listesindeki toplam paket sayısı olan t_s ve $pcktWindow_s$ listesindeki toplam SYN paket sayısı olan s_s hesaplanır. Ardından, kötü amaçlı etkinlik tespiti için birkaç karşılaştırma işlemi gerçekleştirilir. İlk olarak, $pcktWindow_s$ listesindeki ve *windowSize* kapsamındaki toplam paket sayısının oranı hesaplanır ve gelişmemiş kararlardan kaçınmak için bir

Tablo 3. SYN flood tespit algoritması (SYN flood attack detection algorithms)

Satır No	Algoritma
1	Yeni paket (p) girişi
2	s <= Paketin IP adresi kaynağı
3	i <= Paketin giriş anahtarı bağlantı noktası
4	Eğer i bir ana bilgisayar bağlantı noktası ise,
5	$pcktWindow_s[lastEntry_s]$ <= p
6	Eğer $lastEntry_s$ < *windowSize* ise,
7	$lastEntry_s$ artar.
8	veya
9	$lastEntry_s$ <= 0
10	t_s <= $pcktWindow_s$ listesindeki toplam paket sayısı
11	s_s <= $pcktWindow_s$ listesindeki toplam SYN paket sayısı
12	Eğer (s_s / t_s > *synFloodThreshold*) ve (t_s / *windowSize* < *saturationThreshold*)
13	p paketi kötü amaçlıdır
14	t_s <= 0
15	s_s <= 0

Tablo 4. Flood saldırısı tespit algoritması (Flood attack detection algorithms)

Satır No	Algoritma
1	Yeni paket (p) girişi
2	s <= Paketin IP adresi kaynağı
3	d <= Paketin IP adresi hedefi
4	i <= Paketin giriş anahtarı bağlantı noktası
5	*Prop* <= p özelliği
6	$diffDestAddrList_s$ <= farklı hedef IP adreslerinin listesi
7	Eğer i bir ana bilgisayar bağlantı noktası ise,
8	Eğer (d, $diffDestAddrList_s$ ve $diffSrc_s$ içerisinde) ise
9	depolanır.
10	$pcktWindow_s[lastEntry_s]$ <= p
11	Eğer $lastEntry_s$ < *windowSize* ise,
12	$lastEntry_s$ artar.
13	veya
14	$lastEntry_s$ <= 0
15	t_s <= $pcktWindow_s$ listesindeki toplam paket sayısı
16	$occProp_{props}$ <= $pcktWindow_s$ listesindeki toplam kaynak IP adresi paket sayısı
17	$occDest_d$<= $pcktWindow_s$ listesindeki hedefi d olan paket sayısı
18	Eğer (t_s / *windowSize* > *saturationThreshold*) biçimdeyse
19	Eğer (o$occDest_d$ / t_s > *floodThreshold* ve $occProp_{props}$ / t_s > *floodThreshold) ise*
20	p paketi kötü amaçlıdır
21	t_s <= 0
22	o$occDest_d$ <= 0
23	$occProp_{props}$ <= 0

saturationThreshold ile karşılaştırılır. Yeterli sayıda paket varsa, SYN paketlerinin sayısının ve $pcktWindow_s$ listesindeki toplam paket sayısının oranı *synFloodThreshold* değeriyle karşılaştırılarak SYN paketlerinin sayısında çok büyük bir artış olup olmadığı kontrol edilir. Bu durumda sorumlu kaynak IP adresine yönelik bir ceza süresi için kenar anahtarındaki bir kara liste tablosuna eklenir ve ilişkili ana bilgisayar için şimdiye kadar toplanan tüm istatistikler temizlenir.

5.3. Flood Saldırısı Tespit Algoritmaları
(Flood Attack Detection Algorithms)

Flood saldırı tipinde, kötü amaçlı bir ana bilgisayar, kaynaklarını ve iletişim bant genişliğini tüketmek için kurban ana bilgisayara doğru çok sayıda yedekli paket üretir ve gönderir. Çalışmamızda İnternet Denetim İleti Protokolü (Internet Control Message Protocol-ICMP) Flooding ve Kullanıcı Veri Bloğu Protokolü (User Datagram Protocol-UDP) Flooding saldırı tipi örnekleri ele alınmış ve Tablo 4'te açıklandığı gibi ICMP Flooding ve UDP Flooding saldırılarına karşı bir algılama algoritması tasarlanmıştır.

Flood saldırılarını tespit algoritmalarında bir *p* paketi, ana bilgisayar bağlantı noktasından bir kenar anahtarına ulaştığında, $pckWindow_s$ listesinde $lastEntry_s$ dizin numarasıyla depolanır. Listelemede farklı kaynak ana bilgisayarlar için farklı istatistikler ve listeler kullanılır. Paketi listede sakladıktan sonra, $lastEntry_s$ değeri, *windowSize*'ın değerinden küçükse artırma uygulanır ve bununla birlikte $pckWindow_s$ listesindeki paket sayısının ve *windowSize*'ın oranı hesaplanır ve *saturationThreshold* ile karşılaştırma yapılır. Listede yeterli sayıda paket varsa, aynı hedef adrese sahip liste paketlerinin sayısı *p* $occDest_d$ olarak hesaplanır. Ek olarak, *p* ile aynı özelliklere sahip liste paketlerinin sayısı da $occProp_{props}$ şeklinde bulunur. Bu işlemlerin ardından $pckWindow_s$ listesindeki o$occDest_d$ ve toplam paket sayısına oranı hesaplanır. Hesaplanan oran, o kaynak gelen paketlerin hedef IP adreslerinin dağılımını kontrol etmek için *floodThreshold* değeri ile karşılaştırılır. Daha sonra $occProp_{props}$ değerinin $pckWindow_s$ listesindeki toplam paket sayısına oranı bulunarak ana bilgisayardan alınan paketlerin özelliklerinin dağılımı hesaplanır. Bu oran *floodThreshold* değeri ile karşılaştırılır. Karşılaştırma sonuçları, ana bilgisayardan taşma saldırısı olduğunu gösteriyorsa, ana bilgisayarın kaynak IP adresi kara liste tablosunda saklanır ve izlenen özellikler varsayılan değerlerine sıfırlanır ve izleme işlemi yeniden başlatılır.

5.4. Amplifikasyon Saldırı Tespit Algoritmaları
(Amplification Attack Detection Algorithms)

Amplifikasyon saldırı türünde, kötü amaçlı bir ana bilgisayar, kötü amaçlı trafik oluşturur ve bu trafikteki paketlerin bazı özelliklerini, kötü amaçlı etkinliğin etkilerini artırmak için değiştirir. Saldırgan bu saldırıyı kullanarak, saldırıyı oluşturmak için kurban ana bilgisayarın kendi bant genişliğinden daha fazla ağ bant genişliğini kullanır.

Bu çalışmada Broadcast Amplifikasyon Saldırıları, amplifikasyon tipi saldırıların bir örneği olarak ele alınmıştır. Kötü niyetli bir kullanıcı, kötü amaçlı mesajları kurban sunucuya yönlendirmek yerine ağ üzerinden yayınlar veya kötü niyetli bir kullanıcı, reflektörlere yansıma paketleri göndererek yanıtlarını yayınlamasını sağlar. Bu saldırının temel amacı, saldırının altta yatan ağ ve yasal kullanıcılar üzerindeki etkilerini artırmaktır. ICMP isteği gibi bir istek mesajı ağ üzerinden yayınlanırsa ve yasal kullanıcılar buna yanıt verirse, saldırının etkisi yıkıcı olur. MiddleModule sisteminde, yayın yükseltme saldırılarını önlemek için, Tablo 5'te açıklandığı gibi hem yayın mesajlarının sayısını hem de paket özelliklerinin değişimini kontrol eden bir algoritma oluşturulmuştur.

Bu algılama algoritmasının çalışması ise; bir paket ana bilgisayar bağlantı noktasından bir kenar anahtarına ulaştığında, $pckWindow_s$ listesinde $lastEntry_s$ dizin numarasıyla depolanır. $pckWindow_s$ Listesi ve *windowSize* paketlerinin toplam sayısının oranı hesaplanır ve bir *saturationThreshold* ile karşılaştırılır. Eğer $pckWindow_s$ listesinde doğru karar vermeye yetecek kadar paket varsa $pckWindow_s$listesindeki yayın mesajlarının sayısı bulunur. $pckWindow_s$ Listesindeki paket özellikleri, paket bayt uzunlukları ve paket protokolleri de kontrol edilir ve farklı paket özelliklerinin sayısı hesaplanır. Daha sonra, yayın mesajlarının sayısı ve listedeki farklı paket özelliklerinin sayısı, listedeki toplam paket sayısıyla karşılaştırılır. Bu karşılaştırmalar bir yayın yükseltme saldırısı olduğunu gösteriyorsa, önce sorumlu kaynak ana bilgisayarın kaynak IP adresi kara liste tablosunda saklanır ve bu kaynağın tekrar kötü amaçlı olduğu tespit edilirse, kötü amaçlı listede saklanır. Bu algoritmada, kaynak tabanlı akışlar oluşturulur ve kaynak tabanlı her akış için paket izleme ve kötü amaçlı algılama ele alınır.

5.5. Yansıma saldırısı tespit algoritmaları
(Reflection attack detection algorithms)

Yansıma Saldırısı kapsamında, kötü amaçlı bir ana bilgisayar, değiştirilmiş kaynak IP adresiyle reflektörlere birkaç istek mesajı gönderir. Kötü niyetli kullanıcı kendi IP adresini yazmak yerine, mesajların kaynak IP adresi kısmına kurban ana bilgisayarın IP adresini yazar ve buna bağlı olarak yansıtıcılardan gelen yanıtlar kurban ana bilgisayara gönderilir. Bu çalışmada ICMP Yansıma Saldırısı, yansıma saldırı tipinin bir örneği olarak ele alınmış ve Tablo 6'da açıklandığı gibi ICMP Yansıma saldırısına karşı bir algılama algoritması gerçekleştirilmiştir. Yansıma algılama algoritmasında, *p paketi* kenar anahtarına ulaştığında, gelen bağlantı noktası *i*'nin bir ağ bağlantı noktası olup olmadığı kontrol edilir ve eğer öyleyse, pakete yansıma saldırısı algılama algoritması uygulanır. Bu algoritmada, kaynak adres tabanlı akışların daha önce olduğu önceki algoritmalardan farklı olarak hedef adres tabanlı akışlar üretilir. Bir ağ portundan *p paketini* aldıktan sonra, paketi $pckWindow_d$ listesine kaydetmeden önce mevcut $pckWindow_d$ listesi kontrol edilir ve $pckWindow_d$listesindeki paketlerin farklı kaynak IP adresleri$diffSrc_d$ listesinde saklanır. $diffSrc_d$ listesinde yer alan bu paketlerin özellikleri $diffSrcProp_d$ listesinde saklanır. *P paketi* daha sonra $LastEntry_d$ indeksi kullanılarak $pckWindow_d$ listesinde saklanır ve $LastEntry_d$, *windowSize*'dan küçükse artırılır. Daha sonra alınan paketin kaynak IP adresi olan *s*, $diffSrc_d$ listesinde bulunan IP adresleri ile karşılaştırılır ve bu listede yoksa $diffSrc_d$ listesine *s* eklenir ve alınan paketin özelliği $diffSrcProp_d$ listesine eklenir. $pckWindow_d$ listesindeki toplam paket sayısının *windowSize*'a oranı *saturationThreshold* değeri ile karşılaştırılır. $diffSrcProp_d$ listesinde yeterli paket varsa $diffSrc_d$ listesinin boyutunun $pckWindow_d$ listesindeki toplam paket sayısına oranı hesaplanır ve bu oran *refSrcThreshold* ile karşılaştırılır. Eşikten büyükse, hedef ana bilgisayara gönderilen farklı kaynak IP adreslerine sahip çok fazla paket olduğu anlamına gelir. Son olarak, $diffSrcProp_d$ listesindeki paket özellikleri analiz edilir ve diğerleri ile aynı özelliklere sahip olan paketlerin sayısı bulunur; $diffSrcProp_d$ listesindeki aynı özelliklere sahip maksimum paket sayısı $maxNumbSameProp_d$ değerine atanır. $maxNumbSameProp_d$ değerinin $diffSrc_d$ listesinin boyutuna oranı *refPropThreshold*' değerinden büyükse kaynak adresleri farklı ama özellikleri birbirine benzeyen çok fazla paket bulunduğu anlamına gelir. Bu nedenle, bu koşulları sağlayan tüm paketler kötü amaçlı olarak kabul edilir ve bu paketlerin kaynak IP adresleri, denetleyici üzerinden diğer uç anahtarlara gönderilir. Böylece bu ana bilgisayarlardan gelen paketler kaynak tarafında hafifletilir.

6. Sistemin Performans Analizleri (System Performance Analysis)

Önerilen MiddleModule sisteminin değerlendirme sürecinde farklı test platformlarında sisteme çeşitli testler uygulanmıştır. Bu amaçla OMNET++ 4.2 [34] simülasyon platformu ve Mininet [35] emülasyon platformu kullanılmaktadır. Bu platformlarda gerçekleştirilen testler aşağıda sunulmuştur.

Tablo 5. Amplifikasyon saldırı tespit algoritması (Amplification attack detection algorithms)

Satır No	Algoritma
1	Yeni paket (*p*) girişi
2	*s* <= Paketin IP adresi kaynağı
3	*i* <= Paketin giriş anahtarı bağlantı noktası
4	*Prop* <= p özelliği
5	Eğer *i* bir ana bilgisayar bağlantı noktası ise,
6	$pcktWindow_s[lastEntry_s] <= p$
7	Eğer $lastEntry_s$ < *windowSize* ise,
8	$lastEntry_s$ artar.
9	veya
10	$lastEntry_s <= 0$
11	t_s <= $pcktWindow_s$ listesindeki toplam paket sayısı
12	b_s <= $pcktWindow_s$ listesindeki toplam yayın mesajı sayısı
13	$occProp_{props}$ <= $pcktWindow_s$ listesindeki prop özelliğine sahip paket sayısı
14	Eğer (t_s / *windowSize* > *saturationThreshold*) biçimdeyse
15	Eğer (b_s/ t_s> *amplificationThreshold* ve $occProp_{props}$ / t_s > *amplificationThreshold) ise*
16	*p* paketi kötü amaçlıdır
17	t_s<= 0
18	b_s <= 0

Tablo 6. Yansıma saldırısı tespit algoritması (Reflection attack detection algorithms)

Satır No	Algoritma
1	Yeni paket (*p*) girişi
2	*i* <= Paketin giriş anahtarı bağlantı noktası
3	*d*<= Paketin hedef IP adresi
4	*s* <= Paketin IP adresi kaynağı
5	*Prop* <= p özelliği
6	Eğer *i* bir hedef bir port ise,
7	$diffSrc_d$<=$pcktWindow_d$ setindeki paketlerin farklı kaynak IP adresleri
8	$diffSrcProp_d$ <= $diffSrc_d$ setini oluşturan paketlerin özellikleri
9	$pcktWindow_d[lastEntry_d] <= p$
10	Eğer $lastEntry_d$ < *windowSize* ise,
11	$lastEntry_d$ artar.
12	veya
13	$lastEntry_d <= 0$
14	t_d <= $pcktWindow_s$ listesindeki toplam paket sayısı
15	Eğer *s* $diffSrc_d$ değil ise,
16	$diffSrc_d$ İçerisine *s* eklenir,
17	$diffSrcProp_d$ İçerisine prop eklenir.
18	$maxNumbSameProp_d$ <= $diffSrcProp_d$ setindeki aynı özelliklerin maksimum sayısı
19	$samePropSrc_d$ <= $maxNumbSameProp_d$ oluşturan paketlerin kaynak IP adresleri
20	Eğer t_d / *windowSize* > *saturationThreshold* ise
21	Eğer $diffSrc_d$ / t_d > *refSrcThreshold* ise
22	Eğer $maxNumbSameProp_d$/ $diffSrc_d$> *refPropThreshold* ise
23	$samePropSrc_d$ paketi kötü amaçlıdır.
24	$samePropSrc_d$ kontrol listesine gönderilmektedir.
25	t_d<= 0
26	$diffSrc_d$ seti temizlenmekte,
27	$diffSrcProp_d$ seti temizlenmekte,
28	$samePropSrc_d$ seti temizlenmekte,
29	$maxNumbSameProp_d$ <= 0

6.1. OMNET Kapsamında Değerlendirme
(Evaluation within the Scope of OMNET)

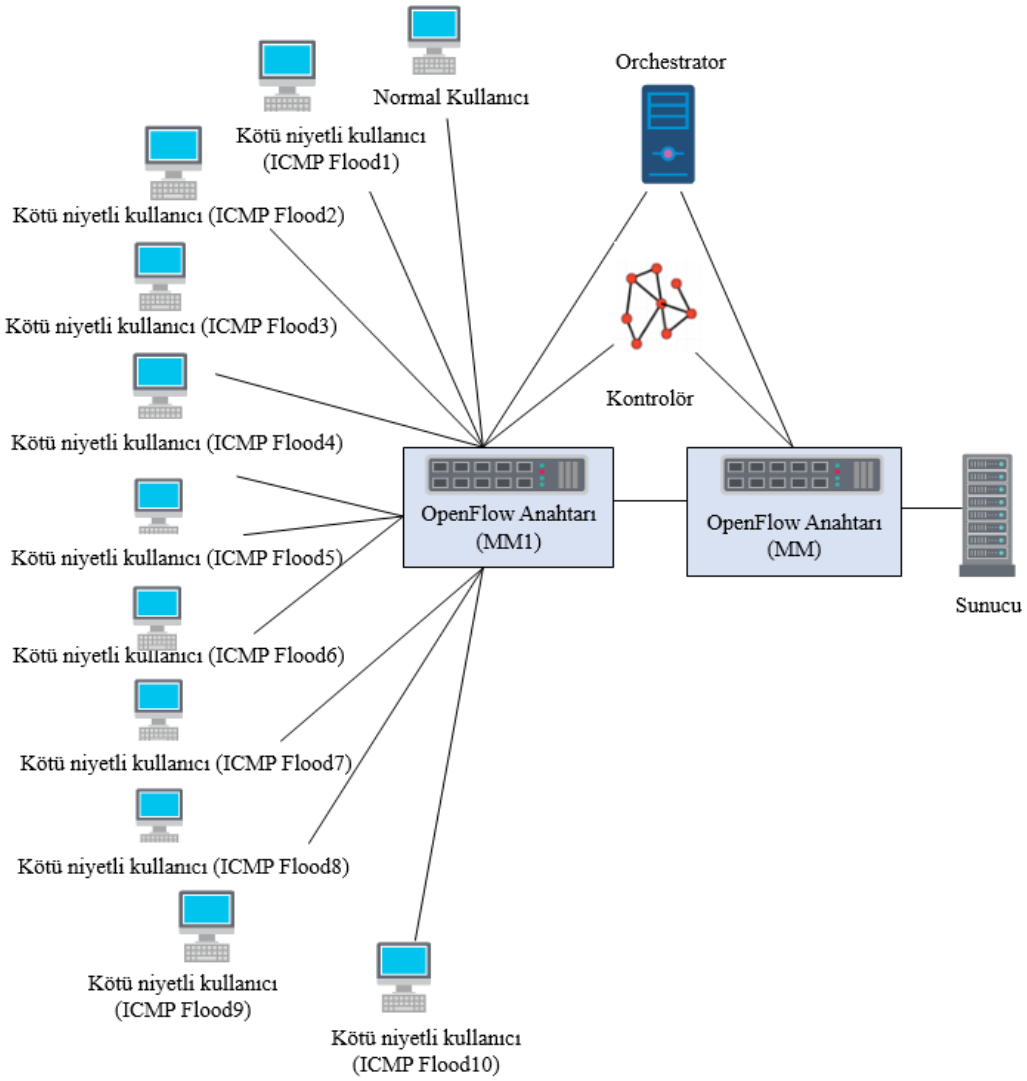
Bu çalışmada; sistem performansında bozulma testi ve savunma gücü testi OMNET++ 4.2 platformu üzerinde Şekil 3'te gösterilen test senaryosu ile gerçekleştirilmiştir.

Sistem performansında bozulma testi kapsamında MiddleModule sistemine yönelik olarak normal kullanıcının ortalama paket alma süresi üzerindeki etkisi ölçülmüştür. Testler, dört farklı senaryo kullanılarak uygulanmıştır: 1) MiddleModule devredeyken normal trafik, 2) MiddleModule devrede değilken normal trafik, 3) MiddleModule devredeyken kötü amaçlı trafik, 4) MiddleModule devrede değilken kötü amaçlı trafik. İlk olarak normal kullanıcı, NSL-KDD veri setini kullanarak sunucuya doğru ve yasal TCP trafiği oluştururken, on tane kötü amaçlı kullanıcıya yönelik ana bilgisayarlar herhangi bir trafik oluşturmamış ve MiddleModule sistemi devre dışı bırakılmıştır. NSL-KDD veri seti yasal trafiği ve kötü amaçlı trafiğin bir kısmını oluşturmak için kullanılmaktadır [36]. Bu veri seti, DDoS saldırılarında ve zararlı trafik verilerini içermesi sebebiyle ve yaygın kullanılmasından dolayı tercih edilmiştir.

Ayrıca, OMNET++ 4.2 simülasyon platformunda, MiddleModule sisteminin duyarlılığı, yanlış pozitif oranı ve doğruluğunu ölçmek amacıyla savunma gücü testleri uygulanmıştır. Savunma gücü testinde, MiddleModule sisteminin kötü amaçlı etkinlikleri tespit edip etmediğini gözlemlemek için temel fonksiyonel analize yönelik testler uygulanmıştır.

Tablo 7. Savunma gücü performans sonuçları (Defence force performance results)

	Duyarlılık	Yanlış Pozitif Oran	Doğruluk Oranı
IP Kandırma Saldırısı	% 99,8	%0	%99,9
SYN Flood Saldırısı	%99,6	%0	%99,8
UDP Flood Saldırısı	%99,6	%0	%99,8
ICMP Yansıma Saldırısı	%99,7	%0	%99,8
Yayın Yükseltme Saldırısı	%99,7	%0	%99,8



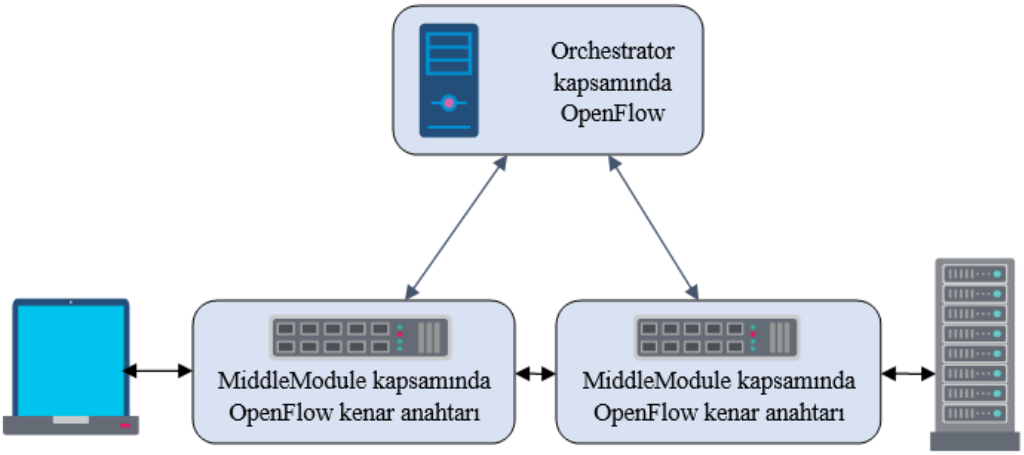
Şekil 3. OMNET++ Kapsamında Sistem Performansı Düşürme Testi Senaryosu (OMNET++ System Performance Decrease Test Scenario)

Gerçekleştirilen testlerin sonucu Tablo 7'de sunulmuştur. Duyarlılık sonuçları tüm saldırı türleri için %99,6'nın üzerinde ve doğruluk sonuçları %99,8'in üzerindedir; bu, hedeflenen tüm Ağ/Aktarım katmanında DDoS saldırı türlerinin MiddleModule sistemi tarafından doğru bir şekilde algılandığı anlamına gelmektedir. Öte yandan, yasal paketler MiddleModule sistemi tarafından bırakılmaz; bu nedenle yanlış pozitif oranı sıfırdır.

6.2. Mininet Kapsamında Değerlendirme
(Evaluation within the Scope of Mininet)

Mininet platformunda Şekil 4'te gösterilen test senaryosu oluşturulmuş ve Sunucu düğümünde bir Hiper Metin Transfer Protokolü (Hyper Text Transfer Protocol-http) web sunucusu uygulaması çalıştırılmıştır. Öncelikle MiddleModule devre dışı bırakılmıştır. Bu durumda Normal Kullanıcı, Sunucuya bağlanarak *http* web sunucusundan paket alır ve ortalama paket alma süresi ölçülür. Bu test, her biri 100 paket geri getirme içeren 10 tur için uygulanmış ve her turda testin tamamı için ortalama paket alma süresi hesaplanmıştır. Ardından MiddleModule etkinleştirilmiş ve Normal Kullanıcı Sunucuya bağlanmıştır. Paket alma süreleri yine 10 tur kapsamında her tur için 100 paket alımı için ölçülmüştür. Normal Kullanıcının ortalama paket alma süresi, bu koşullar altında her tur için hesaplanmıştır.

Mininet platformu üzerinde gerçekleştirilen sistem performans düşüş testi sonuçları Tablo 8'de gösterilmiştir. SDN mimarisi içerisinde MiddleModule sistemin olup olmamasına göre testler yapılmıştır.



Şekil 4. Mininet Açısından Sistem Performans Düşüşü Test Senaryosu (Mininet System Performance Decrease Test Scenario)

Tablo 8. Sistem performansı düşüşü (System performance degradation)

Tur	MiddleModule yok (saniye)	MiddleModule var (saniye)
Tur 1	0.02530	0.02532
Tur 2	0.02516	0.02531
Tur 3	0.02531	0.02526
Tur 4	0.02542	0.02544
Tur 5	0.02534	0.02532
Tur 6	0.02542	0.02528
Tur 7	0.02528	0.02556
Tur 8	0.02501	0.02547
Tur 9	0.02533	0.02503
Tur 10	0.02556	0.02520
Ortalama	0.025313	0.025319
Genel	-	% 0,2

Elde edilen sonuçlara göre saniye cinsinden ortalama paket alma sürelerinin birbirine çok yakın olduğu ve MiddleModule sisteminin neden olduğu paket alma süresi ek yükünün %0,2 olduğu belirlenmiştir. Bu nedenle, paket alma süresi açısından MiddleModule sisteminin sistem performansı düşüşü ihmal edilebilir seviyededir.

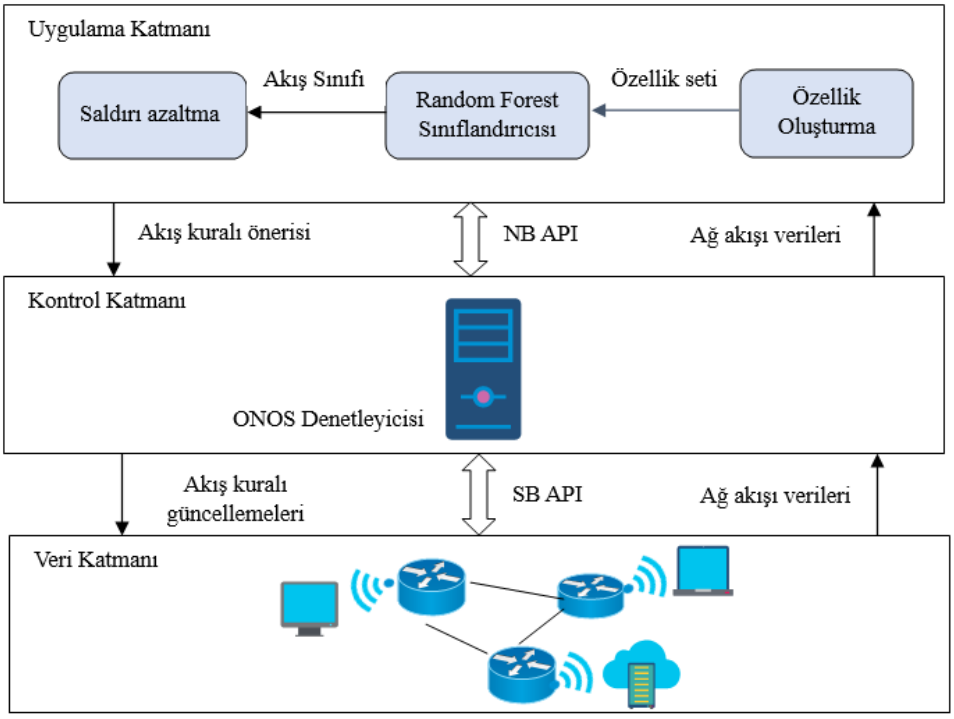
6.3. Önerilen Güvenlik Yaklaşımları
(Recommended Security Approaches)

Bu çalışmada önerilen SDN Tabanlı Güvenlik Çözümü Mimarisi'nin (Şekil 5) genel işleyişi, izinsiz girişi tespit etme ve azaltma yaklaşımı, ağ akışlarının otomatikleştirilmiş, akıllı analizi ve ardından izinsiz giriş tespit bileşeninin kararına uygun olarak alınan hafifletme eylemleriyle SDN tabanlı ağlarda güvenlik sağlamaktadır. Uçtan uca saldırı tespiti ve azaltma işlemi, uygulama katmanında bir SDN uygulamasında çalışan; Özellik Oluşturma, RF sınıflandırıcı ve Saldırı Azaltma üç ana modüle dayanır.

Özellik Oluşturma Modülü, anahtarlardan düzenli aralıklarla ağ akışlarını toplar ve her akış için RF sınıflandırıcının gerektirdiği özellikleri hesaplar. RF sınıflandırıcı, önceden oluşturulmuş izinsiz giriş algılama modelini akış örneğine uygular ve sonucu Saldırı Azaltma Modülüne iletir. Saldırı Azaltma Modülü daha sonra sınıflandırma sonucuna göre yapılacak eylemi belirler ve gerekirse saldırıyı azaltmak için ilgili anahtarlara akış kuralları yükler. Tablo 9'da yer alan algoritma, önerilen güvenlik çözümünün uçtan uca çalışmasını özetlemek maksadı ile verilmiştir. Denetleyici, düzenli aralıklarla Özellik Oluşturma Modülü tarafından alınan anahtarlardan ağ akış girişlerini düzenli olarak toplar. Sonrasında, her akış için özellikler oluşturulur. Her akış için ortak özellikler, anahtardaki her akış girişi üzerinden döngü halinde oluşturulur. Örneğin; ortalama akış süresi ve işlemdeki toplam paket sayısı, ilk geçiş akışı girişleriyle oluşturulur. Ardından, akışa özgü özellikler, örneğin akış süresi ve kaynaktan hedefe paket sayısı, tüm akış girişlerinin üzerinden geçilerek alınır. Akış girişleri üzerinde döngü yaparken, bir akış için oluşturulan özellik vektörü, diğer akış girişleri için özellik oluşturmayı bitirmeyi beklemeden hemen RF sınıflandırıcıya gönderilir. Özellik Oluşturma Modülü, kaynak IP ve MAC adresleri gibi akış eşleştirme alanlarını ve paketin geldiği anahtarın fiziksel bağlantı noktasını da alır. Bu özellikler Saldırı Azaltma Modülü tarafından kullanılır.

6.4. Önerilen MiddleModule Sisteminin Karşılaştırılması
(Comparison of the Proposed MiddleModule System)

Önerilen MiddleModule sistemin performans değerlendirmesi literatürde mevcut bulunan veri katmanı tabanlı Dağıtılmış Hizmet Reddi savunma sistemleri ile kıyaslandığında oldukça iyi sonuçlar elde edilmektedir. Önerilen MiddleModule sistemimiz ile AVANT-GUARD, CIDS, StateSec ve SDNScore savunma sistemleri kıyaslanmıştır. MiddleModule sistemi ve tüm çalışmalar Dağıtılmış Hizmet Reddi saldırılarına karşı önerilmiş olsa da yukarıdaki çalışmalarda hedeflenen saldırı türleri birbirinden biraz farklıdır. AVANT-GUARD sistemi sadece SYN Flood Dağıtılmış Hizmet Reddi saldırı türünü dikkate alır, CIDS sistemi SYN Flood saldırılarını ve Flooding saldırılarını (ICMP Flooding ve UDP Flooding) dikkate alır, StateSec sistemi ise sadece Flooding saldırılarına karşı test edilir.



Şekil 5. SDN tabanlı güvenlik çözümü mimarisi (SDN based security solution architecture)

Tablo 9. Uçtan uca işlem adımları (End-to-end process steps)

Satır No	Algoritma
1	L_S← Bağlı olan SDN anahtarları alınır.
2	M←RF sınıflandırıcı modeli yüklenir.
3	L_B← Kara Liste
4	Doğru ise,
5	S ve L_S için
6	L_E←S için akış girişleri çekilir.
7	L_E=Özellik_Oluşturma
8	Son.
9	Bir süre beklenir.
10	Son.
11	Özellik Oluşturma İşlevi:
12	F_1← L_E ile ortak özellikleri belirlenir.
13	E ve L_E için
14	F_2← E için akışa özgü özellikler hesaplanır.
15	F←(F_1.F_2)
16	Saldırı Tespiti (F)
17	Son.
18	Saldırı Tespiti İşlevi (Özellik Vektörü F):
19	C← F ve M sınıflandırmak için kullanılır.
20	Eğer C saldırı ise,
21	I ← Kaynak tanımlayıcılarını F den almaktadır.
22	Azaltma (C.I)
23	Son.
24	Azaltma İşlevi (C *saldırı tipi*, I *tanımlayıcı*) :
25	Eğer L_B içerisinde I bulunmuyorsa,
26	E← $I'nın$ engellenmesi ya da yeniden yönlendirilmesi için akış girişi oluşturulmaktadır.
27	E, $S'ye$ takılır.
28	I, L_B içerisine eklenir.
29	Son.

Tablo 10. İdeal koşullar altında savunma gücü karşılaştırması (Defense force comparison under ideal conditions)

	ICMP veya UDP Flooding Saldırıları		SYN Flood Saldırıları	
Sistem	Yanlış Pozitif Oran	Duyarlılık	Yanlış Pozitif Oran	Duyarlılık
MiddleModule	%0	%99,3	%0	%99,3
SDNScore	%1	%100	%0,02	%100
StateSec	%0	%100	-	-
CIDS	%3.4	%96,3	%2,3	%94,8

Tablo 11. Paket Alma Zaman Gecikmesi Karşılaştırması (Packet Receiving Time Delay Comparison)

Karşılaştırma Parametresi	MiddleModule	AVANT-GUARD
Paket alma süresinde gecikme oranı	%0,2	%1,86

Tablo 12. Trafik oluşturma oranlarının karşılaştırması (Comparison of traffic generation rates)

	En az paket üretim hızı	En fazla paket üretim hızı
MiddleModule	10 ms(100 adet/saniye)	2 μs (5.000.000 adet/saniye)
StateSec	10 ms (100 adet/saniye)	1200 μs (833 adet/saniye)
CIDS	1 ms (100 adet/saniye)	200 μs (5.000 adet/saniye)
AVANT-GUARD	10 ms (100 adet/saniye)	1 ms (1.000 adet/saniye)

SDNScore, SYN Flood saldırısı ve Flooding saldırılarına odaklanır. MiddleModule sistemi ise IP Kandırma saldırısı, SYN Flood saldırısı, kandırma Flooding saldırısı, kandırma olmadan Flooding saldırısı, Yansıma saldırısı ve Broadcast Amplification saldırılarına karşı tasarlanmış ve test edilmiştir. Bu nedenle, MiddleModule sistemi daha geniş bir Ağ/Aktarım katmanında DDoS saldırı setini kapsamaktadır. Çalışmaların savunma güçleri önerdiğimiz yöntemle karşılaştırılmıştır. Bu amaçla hem ideal hem de ideal olmayan koşullar altında kıyaslama yapılmaktadır. İdeal koşul, saldırıların iyi tanımlanmış paket özniteliklerine (paket uzunlukları, protokoller vb.) sahip olduğu ve normal hızda üretildiği anlamına gelir. Tablo 10'da ideal koşullar altında savunma gücü karşılaştırması sonuçları verilmiştir. AVANT-GUARD sistemi için verilen sayısal savunma gücü değerleri olmadığı için savunma gücü karşılaştırmasında kullanılmamaktadır. Önerilen MiddleModule savunma sistemimizin farklı DDoS saldırı türlerinde oldukça iyi sonuçlar verdiği görülmektedir. Bununla birlikte, ideal olmayan koşullarda, MiddleModule sisteminin savunma gücü performansı diğer çalışmalara göre belirgin şekilde yüksektir. İdeal olmayan durumda diğer sistemlerinin algılama oranı yaklaşık %80'e düşerken, MiddleModule sisteminin algılama oranının ideal koşullarda toplanan sonuçlara yakın olan %96'nın üzerinde kalmaktadır.

Paket alma süresinde gecikme oranları karşılaştırıldığında MiddleModule ve AVANT-GUARD sistemleri için elde edilen oranlar Tablo 11'de verilmiştir. Önerilen sistemimizin %0,2 oranında düşük gecikme süresine sahiptir.

Genel olarak, bir ana bilgisayar ağa Ağ/Aktarım temelli DDoS trafiği oluşturuyorsa mümkün olan en yüksek üretim hızına sahip trafiği üretmektedir. Bu nedenle, yüksek trafik üretim oranları kullanan önerilen savunma sistemlerini değerlendirmek kritik öneme sahiptir. Bir savunma sistemi, yalnızca bugünün ihtiyaçlarından daha düşük paket hızlarında test edilirse, o sistemin savunma gücü, gerçek ağ oluşturma koşullarında anlamsız hale gelebilir. Önerilen MiddleModule sistemi ve diğer savunma sistemlerinin minimum ve maksimum paket üretim hızları Tablo 12'de verilmiştir. Görüldüğü üzere, en az paket üretim hızları birbirine benzer olmakla birlikte maksimum hızlar farklıdır. MiddleModule sisteminde değerlendirme sırasında kullanılan maksimum paket üretim hızı diğer çalışmalara göre yüz kat daha fazladır.

7. Sonuçlar (Conclusions)

Bu çalışmada, Yazılım Tanımlı Ağlar (SDN) ortamında Dağıtılmış Hizmet Reddi (DDoS) saldırılarına karşı etkili bir savunma sistemi geliştirilmesi için veri katmanı tabanlı ve "MiddleModule" olarak adlandırılan bir savunma sistemi önerilmiştir. Önerilen MiddleModule sistemi, DDoS saldırılarının gerçek zamanlı tespitini ve önlenmesini sağlayan hafif ve ölçeklenebilir bir mimariye sahiptir. Sistemin tasarımı aşamasında kullanılan bloklarda; uç anahtarlara izleme, algılama ve önleme işlemleri gerçekleştirilmektedir. Bunun yanı sıra, her bir ağ saldırı türünün en yaygın örneklerine karşı etkili olması beklenen DDoS algılama/tespit algoritmaları tasarlanarak MiddleModule mimarisi içerisinde uygulanmıştır.

Önerilen metot kapsamında farklı test senaryoları altında detaylı bir fonksiyonel analiz ortamı için OMNET ++ platformunda testler gerçekleştirilmiştir. Ayrıca bu simülasyona ek olarak, önerilen sistemin performansını gerçek zamanlı analiz etmek için Mininet ortamında da modellenerek testler yapılmıştır. Testler için NSL-KDD veri seti, akademik çalışmalarda sıklıkla kullanılması ve standard bir test ortamı sağlaması amacıyla tercih edilmiştir. Ayrıca IoT ağları için yeterli ve halka açık veri setleri olmamasından dolayı yeni bir veri seti tasarlanmıştır. Bu veri seti, normal trafik ve farklı saldırı trafiği türlerini içermektedir ve bu sayede sistemin performansı gerçek dünya senaryolarına daha iyi uyum sağlayacak şekilde test edilmektedir.

Önerilen MiddleModule sisteminin ideal olan ve ideal olmayan koşullarda oldukça iyi performans gösterdiği görülmüştür. İdeal olmayan koşullarda diğer savunma sistemlerinin gücünün önemli ölçüde azaldığı görülmektedir. Sistem performans düşüşü, özellikle veri katmanı tabanlı savunma sistemleri için önemli bir husustur, çünkü bu tür sistemlerde ağdaki tüm paketler, savunma sisteminin neden olduğu performans düşüşünü yaşar. Ancak önerilen MiddleModule sistemi test sonuçların yer aldığı Tablo 8'de görüldüğü üzere ağ üzerindeki ek yükü %0,2 seviyesine indirerek sistem performansı üzerindeki olumsuz etkilerini ihmal edilebilir düzeyde seviyededir. Bu performans kriteri de göz önünde bulundurularak, MiddleModule; SDN denetleyicisini koruma altına alarak kötü amaçlı trafiğin veri katmanında tespit edilmesini ve önlenmesini sağlamaktadır. Bu tasarım, ağ tıkanıklığını ve denetleyiciye yönelik saldırı riskini önemli ölçüde azaltmaktadır. Bununla birlikte hem ideal hem de ideal olmayan koşullarda savunma gücü karşılaştırıldığında

önerilen MiddleModule sistemi yanlış pozitif oranı, duyarlılık ve paket alma sürelerinde gecikme ve trafik oluşturmada veri katmanı tabanlı diğer savunma sistemlerine kıyasla oldukça iyi sonuçlar verdiği Tablo 10, Tablo 11 ve Tablo 12'de gösterilmiştir.

Sonuç olarak; gerçek zamanlı saldırılar açısından bu çalışma, 5G çağının IoT ağlarında açıklanabilir güvenlik sağlamayı amaçlayan SDN'lar için otomatik, akıllı bir saldırı tespiti ve azaltma yaklaşımı sunulmaktadır. Bununla birlikte; Önerilen sistem veri katmanında saldırıları hızlı bir şekilde tespit ederek kontrol katmanının yükünü azaltmaktadır. Saldırı tespit/ algılama için kullanılan hafif algoritmalar sayesinde, farklı ağ topolojilerinde ve yüksek trafik koşullarında dahi önerilen sistem etkinliğini koruyabilmektedir. Farklı DDoS saldırı türlerini kapsayan bir tasarım sunarak farklı türdeki saldırılara karşı SDN'i korumaktadır. Bununla birlikte; çalışmada kullanılan veri setlerinin gerçek dünyadaki tüm siber saldırı senaryolarını kapsamamaktadır. Gelişmiş ağ topolojileri ve IoT cihazlarının neden olduğu yeni saldırı türleri için önerilen sistemin daha fazla optimize edilmesi ve farklı saldırı türleri için de algoritma tasarlanması gerekebilir.

Kaynaklar (References)

1. Zargar S.T., Joshi J., Tipper D., A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials, 15 (4), 2046-2069, 2013.
2. Owusu E., Rahouti M., Jagatheesaperumal S. K., Xiong K., Xin Y., Lu L., Online Network DoS/DDoS Detection: Sampling, Change Point Detection, and Machine Learning Methods. IEEE Communications Surveys & Tutorials, 2024.
3. Yan Q., Yu F. R., Gong Q., Li J., Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Communications Surveys & Tutorials, 18 (1), 602-622, 2015.
4. Kaur A., Krishna C. R., Patil N. V., K-DDoS-SDN: A distributed DDoS attacks detection approach for protecting SDN environment. Concurrency and Computation: Practice and Experience, 36, 2024.
5. Singh C., Jain A. K., A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. e-Prime - Advances in Electrical Engineering, Electronics and Energy, 8, 2024.
6. Kalkan K., Gür G., Alagöz F., SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment. IEEE Symposium on Computers and Communications (ISCC), 669-675, 2017.
7. Gökrem L., Bozuklu M., Nesnelerin interneti: yapılan çalışmalar ve ülkemizdeki mevcut durum. Gaziosmanpaşa Bilimsel Araştırma Dergisi, (13) 47-68, 2016.
8. Koroniotis N., Moustafa N., Sitnikova E., Turnbull B., Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems, 100, 779-796, 2019.
9. Shin S., Yegneswaran V., Porras P., Gu G., Avant-guard: Scalable and vigilant switch flow management in software-defined networks. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 413-424, 2013.
10. Chen X., Yu S., A collaborative intrusion detection system against DDoS for SDN. IEICE Transactions on Information and Systems, 99 (9), 2395-2399, 2016.
11. Boite J., Nardin P.A., Rebecchi F., Bouet M., Conan V., Statesec: Stateful monitoring for DDoS protection in software defined networks. In 2017 IEEE conference on network softwarization (NetSoft), 1-9, IEEE 2017.
12. Shukla P., Krishna C. R., Patil N. V., Iot traffic-based DDoS attacks detection mechanisms: A comprehensive review. The Journal of Supercomputing, 80, 9986-10043, 2024.
13. Genç Y., Afacan E., Identity-Based Encryption in the Internet of Things, 2021 29th Signal Processing and Communications Applications Conference (SIU) IEEE, 1-4, 2021.
14. Sharma A., Thangaraj V., Intelligent service placement algorithm based on DDQN and prioritized experience replay in IoT-Fog computing environment. Internet of Things, 25, 2024.
15. Rathore, S., and Park, J. H., Semi-supervised learning based distributed attack detection framework for IoT. Applied Soft Computing, 72, 79-89, 2018.
16. Evmorfos, S., Vlachodimitropoulos, G., Bakalos, N., and Gelenbe, E., Neural network architectures for the detection of SYN flood attacks in IoT systems. In Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, 1-4, 2020.
17. Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., and Sakurai, K., Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors, 20 (16), 4372, 2020.
18. Alqahtani, M., Mathkour, H., and Ben Ismail, M. M., IoT botnet attack detection based on optimized extreme gradient boosting and feature selection. Sensors, 20 (21), 6336, 2020.
19. Ravi N., Shalinie S.M., Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture, IEEE Internet of Things Journal, 7 (4), 3559-3570, 2020.
20. Galeano-Brajones, J., Carmona-Murillo, J., Valenzuela-Valdés, J. F., and Luna-Valero, F., Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: An experimental approach. Sensors, 20 (3), 816, 2020.
21. Amangele, P., Reed, M. J., Al-Naday, M., Thomos, N., and Nowak, M., Hierarchical machine learning for IoT anomaly detection in SDN. In 2019 International Conference on Information Technologies (InfoTech) IEEE, 1-4, 2019.
22. Derhab, A., Guerroumi, M., Gumaei, A., Maglaras, L., Ferrag, M. A., Mukherjee, M., and Khan, F. A., Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security. Sensors, 19 (14), 3119, 2019.
23. Wang, M., Zheng, K., Yang, Y., and Wang, X., An explainable machine learning framework for intrusion detection systems. IEEE Access, 8, 73127-73141, 2020.
24. Gündüz S., Sağıroğlu Ş., Design and evaluation of a controller for energy efficient security management and traffic routing in SDN/NFV based 5G networks, Journal of the Faculty of Engineering and Architecture of Gazi University, 37 (1), 1-16, 2022.
25. Tonkal Ö., Polat H., Traffic Classification and Comparative Analysis with Machine Learning Algorithms in Software Defined Networks. GUJ Sci, Part C, 9 (1), 071-083, 2021.
26. Kreutz D., Ramos F.M., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S., Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103 (1), 14-76, 2014.
27. Alharbi T., Layeghy S., Portmann M., Experimental evaluation of the impact of DoS attacks in SDN. In 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), 1-6, 2017.
28. Lim S., Yang S., Kim Y., Yang S., Kim H., Controller scheduling for continued SDN operation under DDoS attacks. Electronics Letters, 51 (16), 1259-1261, 2015.
29. Lau W., Wong K., Cui L., Optimizing the performance of OpenFlow Protocol over QUIC. Journal of Network and Computer Applications, 226, 2024.
30. Mckeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38 (2), 69-74, 2008.
31. Mostafaei H., Menth M., Software-defined wireless sensor networks: A survey. Journal of Network and Computer Applications, 119, 42-56, 2018.
32. Barakabitze A.A., Ahmad A., Mijumbi R., Hines A., 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. Computer Networks, 167, 1-5, 2020.
33. Polat H., Polat O., Cetin A., Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Model. Sustainability, 2020.
34. OMNET ++ 4.2 platformu için erişim adresi: https://omnetpp.org/, 08.01.2025.
35. Mininet platformu için erişim adresi: https://mininet.org/, 08.01.2025.
36. Tavallaee, M. Bagheri, E., Lu, W and Ghorbani, A. A., A detailed analysis of the KDD CUP 99 data set, in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.