DDOS PREDICTION AND MITIGATION IN SDN USING ARTIFICIAL NEURAL NETWORKS AND BMNABC ALGORITHM

2024
MASTER THESIS
ELECTRICAL AND ELECTRONICS ENGINEERING

Esam ATEEYAH

Thesis Advisor
Assist. Prof. Dr. Cihat ŞEKER

DDOS PREDICTION AND MITIGATION IN SDN USING ARTIFICIAL NEURAL NETWORKS AND BMNABC ALGORITHM

Esam ATEEYAH

Thesis Advisor

Assist. Prof. Dr. Cihat ŞEKER

T.C.
Karabuk University
Institute of Graduate Programs
Department of Electrical and Electronics Engineering
Prepared as
Master Thesis

KARABUK
August 2024

I certify that in my opinion the thesis submitted by Esam ATEEYAH titled "DDOS PREDICTION AND MITIGATION IN SDN USING ARTIFICIAL NEURAL NETWORKS AND BMNABC ALGORITHM" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Cihat ŞEKER .....

Thesis Advisor, Department of Electrical and Electronics Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis 02/08/2024

Examining Committee Members (Institutions)	Signature
Chairman : Assist. Prof. Dr. Tarık Adınan ALMOHAMAD (KBU)	.....
Member : Assist. Prof. Dr. Cihat ŞEKER (Izmir Bakircay University)	.....
Member: Assist. Prof. .Dr. Bayram KÖSE (Izmir Bakircay University)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Assoc. Prof. Dr. Zeynep ÖZCAN .....

Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Esam ATEEYAH

ABSTRACT

M.Sc. Thesis

DDOS PREDICTION AND MITIGATION IN SDN USING ARTIFICIAL NEURAL NETWORKS AND BMNABC ALGORITHM

Esam ATEEYAH

Karabük University
Institute of Graduate Programs
Department of Electrical and Electronics Engineering

Thesis Advisor:
Assist. Prof. Dr. Cihat ŞEKER
August 2024, 79 pages

By centralizing control functions and separating control and data planes, Software-Defined Networking (SDN) facilitates improved management and optimization of networks. Yet, this centralization is also a source of vulnerabilities, which are characterized by reasons like Distributed Denial-of-Service (DDoS) attack types that can cripple network resources and impair service delivery systems. This thesis presents novel methods to lessen these attacks within the SDN environments. The aim of this study is to investigate how artificial neural networks (ANNs) can be used in conjunction with the binary multi-neighbor artificial bee colony (BMNABC) algorithm to make security more reliable and resistant to DDoS attacks. The BMNABC algorithm optimizes the allocation of network resources to mitigate the effects of attacks, while ANN detects anomalous traffic patterns that indicate DDoS attacks in real time. The intelligent intrusion detection system (IIDS) uses the "BMNABC" technique to select features and "ANN" to classify data into attack or

non-attack classes. The research additionally makes a comparison between machine learning techniques such as Ensemble, K-Nearest Neighbors (K-NN), Decision Tree (DT), Support Vector Machine (SVM), and Naive Bayes (NB), where the ANN gets the best performance metrics of 98.99% accuracy, 98.97% test accuracy, 98.94% F1 score, 98.95% precision, along with 97.95% recall. DT achieved the lowest metrics: 97.54% accuracy, 97.56% test accuracy, 97.42% F1 score, 97.76% precision, and 97.09% recall.

Keywords: Artificial neural network, DDoS, SDN, Binary multi-neighborhood artificial bee colony algorithm.

Science Code: 915.1.092

ÖZET

Yüksek Lisans Tezi

YAPAY SİNİR AĞLARI ve BMN-ABC ALGORİTMALARI KULLANILARAK YAZILIM TABANLI AĞLARDA DDoS SALDIRILARININ TAHMİNİ ve AZALTILMASI

Esam ATEEYAH

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Cihat ŞEKER
Ağustos 2024, 79 sayfa

Yazılım Tanımlı Ağ İletişimi (SDN), kontrol işlevlerini merkezileştirerek ve kontrol ile veri düzlemlerini ayırarak, ağların gelişmiş yönetimini ve optimizasyonunu kolaylaştırır. Ancak bu merkezileştirme aynı zamanda ağ kaynaklarını felce uğratabilecek ve hizmet dağıtım sistemlerini bozabilecek Dağıtılmış Hizmet Reddi (DDoS) saldırı türleri gibi nedenlerle karakterize edilen bir güvenlik açıklarının da kaynağıdır. Bu tez, SDN ortamlarındaki bu saldırıları azaltmak için yeni yöntemler sunmaktadır. Bu araştırmanın amacı, İkili Çok Komşulu Yapay Arı Kolonisi (BMNABC) algoritmasını Yapay Sinir Ağları (YSA) olarak adlandırılan sistemlerle birleştirerek “DDoS” saldırılarına karşı güvenliğin güvenilirliğini ve sağlamlığını sağlamaktır. "BMNABC" algoritması, saldırıların etkilerini en aza indirecek şekilde ağ kaynaklarının tahsisini optimize ederken "ANN", "DDoS" saldırılarının göstergesi olan anormal trafik modellerini gerçek zamanlı olarak tespit eder. Akıllı izinsiz giriş

tespit sistemi (IIDS), özellikleri seçmek için "BMNABC" tekniğini ve verileri saldırı veya saldırı dışı sınıflara sınıflandırmak için "ANN" tekniğini kullanır. Araştırma ayrıca YSA'nın en iyi sonucu aldığı Ensemble, K-En Yakın Komşular (K-NN), Karar Ağacı (DT), Destek Vektör Makinesi (SVM) ve Naive Bayes (NB) gibi makine öğrenme teknikleri arasında bir karşılaştırma da yapıyor. %98,99 doğruluk, %98,97 test doğruluğu, %98,94 F1 puanı, %98,95 kesinlik ve %97,95 geri çağırma performans ölçümleri. DT en düşük ölçümleri elde etti: %97,54 doğruluk, %97,56 test doğruluğu, %97,42 F1 puanı, %97,76 kesinlik ve %97,09 geri çağırma.

Anahtar Kelimeler : Yapay sinir ağı, DDoS, SDN, İkili çoklu mahalle yapay arı kolonisi algoritması.

Bilim Kodu: 915.1.092

ACKNOWLEDGMENT

First of all, I would like to give thanks to my advisor, Assist. Prof. Dr. Cihat ŞEKER, for his great interest and assistance in preparation of this thesis.

I would like to thank my thesis examiners for their presence in time and their recommendations to upgrade this work.

To my mother, thank you so much mother for your support and your motivation to reach this honor stage, she supported me all the time.

I want to thank my brothers, sisters and my wife for their supporting, and forbearance all the time during my studies.

I hope from any one I forgot to mention him to consent my apology, there are so many good friends deserve the acknowledgment.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVITIONS INDEX

ABC : Artificial Bee Colony

AI : Artificial Intelligence

ANN : Artificial Neural Network

BMNABC : Binary Multi-Neighborhood Artificial Bee Colony

DDoS : Distributed Denial-of-Service

DL : Deep Learning

DNN : Deep Neural Network

ID : Intrusion Detection

IDS : Intrusion Detection System

IID : Intelligent Intrusion Detection System

IoT : Internet of Things

ML : Machine Learning

NIDS : Network Intrusion Detection System

NT : Network Traffic

ONF : Open Networking Foundation

SBN : Software-based Network

SDN : Software-Defined Networking

SNN : Siamese Neural Network

SON : Software-Oriented Network

PART 1

INTRODUCTION

In modern times, the increasing number of interconnected devices and the rapid growth of data traffic have reshaped the network communications environment. Along with these developments, Software-Defined Networking (SDN) has emerged, promising network administrators efficiency, scalability, and flexibility like never before. The core idea of software-defined networking (SDN) is to separate the control and data levels, which has allowed for programmability and centralised control to completely transform network architecture [1], [2].

However, with new innovations come new challenges; Distributed Denial of Service (DDoS) attacks are a big risk to infrastructure, particularly contemporary networks. In order to make resources on the network unavailable to authorised users, these attacks bombard specific systems or services with traffic. Unlike traditional network topologies in which defensive mechanisms are spread all over the network, in Software Defined Networks (SDN), DDoS attackers can use this single point of failure through its central control plane [3].

DDoS attacks can result in significant loss of revenue and can be particularly damaging to a company's reputation, and service outages, among other serious adverse effects on governments, businesses, and key infrastructure. Such complex DDoS attacks cannot be countered by Protection techniques that have been around for a long time, such as firewalls and intrusion detection systems, especially in software-defined networking (SDN) environments, where traffic patterns are usually ever-changing and resource provisioning happens rapidly [4].

To address these challenges, this thesis investigates novel approaches to increasing the resilience of SDN systems against DDoS attacks. We propose combining (ANN) with

(BMNABC) to create a new DDoS defense method. Machine learning (ML) strategies are used by artificial neural networks (ANN) in real-time to examine the behaviors of those networks and identify irregularities that would signify a DDoS attack. Moreover, this algorithm can enhance resource distribution across networks, mitigating the impact of any attacks detected, thereby allowing an efficient management and effective response [5].

This thesis offers a fresh perspective on DDoS mitigation techniques suited to the conditions of software-defined networks, thus contributing to the broadening corpus of research in SDN security. As the world becomes more interconnected, network administrators and security experts who seek to shield essential network infrastructures from cyberattacks will find it of great worth. It is believed that SDN is responsible for the major changes to computer networks [6]. The software-oriented network (SON) can be seen as a distinct structure within the new network due to the inherent logical separation between the control plane and the data within it. One primary goal of SBN architecture was to protect networks from distributed denial of service (DDoS) assaults. Through the utilisation of AI's combined capabilities. Stanford University was an innovator in this area, and other companies followed suit. Originally, packet routing in networks was accomplished using switches known as Ethernet switches. The central processing unit (CPU), sometimes called a control plane, makes the decision to transmit data or not in these sections [7]. Configuring these networks takes a lot of time and effort from administrators.

The Open Networking Foundation (ONF) and other SBN have presented and provided support in order to find the best solution for these problems. The ONF is an organization that builds SON and enhances and extends SDN using open-source criteria. The equipment of the networks such as SON, or SDNs, contains of routers, switches and middlebox appliances, such as the traditional networks. In these types of networks, the control level named the network's intelligence and the data level named the executive arm of this intelligence [7], [8].

Applications work with the network operating system together as an SDN control system [7], [9]. The three are the application layer, control unit layer, and internet

component. They have a three-layer architecture for orientation and data transfer. In this design, the application elements are handled by the top layer, data control is handled by the middle layer, and data is handled by the bottom layer. There are two-ways communication as it's evident, it's utilized to control the data flowing between the application part and the SON or SDN part. The data part includes the data, and the SDN controller part considers as a director for data flowing. Lately metaheuristic techniques has been utilized in several optimization algorithms [10]–[14].

Computer networks have undergone a major shift thanks to (SDN) [6]. The “SON” can be considered a new network design because it allows for the logical separation of the control plane of the network and the data. Stanford University was the first institution to propose the concept of software-based network architecture, which was subsequently developed further by other companies. In the past, Ethernet switches were utilised for the purpose of routing packets within a network structure. The hardware devices in question were referred to as control planes, and they contained a processing unit that made the decision regarding whether or not to transfer data [23]. To configure these kinds of networks, managers are required to devote a significant amount of their time. In order to address these concerns, the “ONF”, which is one of the software-based networks that will be introduced and supported, is one of the organisations that will be involved. The “ONF” is an organisation that aims to develop software-oriented networks by promoting and popularising software-defined networking (SDN) through the utilised open-source standards. Software-defined networks, also known as SDNs, are comprised of a collection of network hardware that is comparable to that of conventional networks. This hardware includes switches, routers, and middlebox appliances. In the context of these kinds of networks, the data plane is also referred to as the executive arm of this intelligence, whereas the control level is referred to as the intelligence of the network. Figure 1.1 [7] depicts the SDN network architecture.

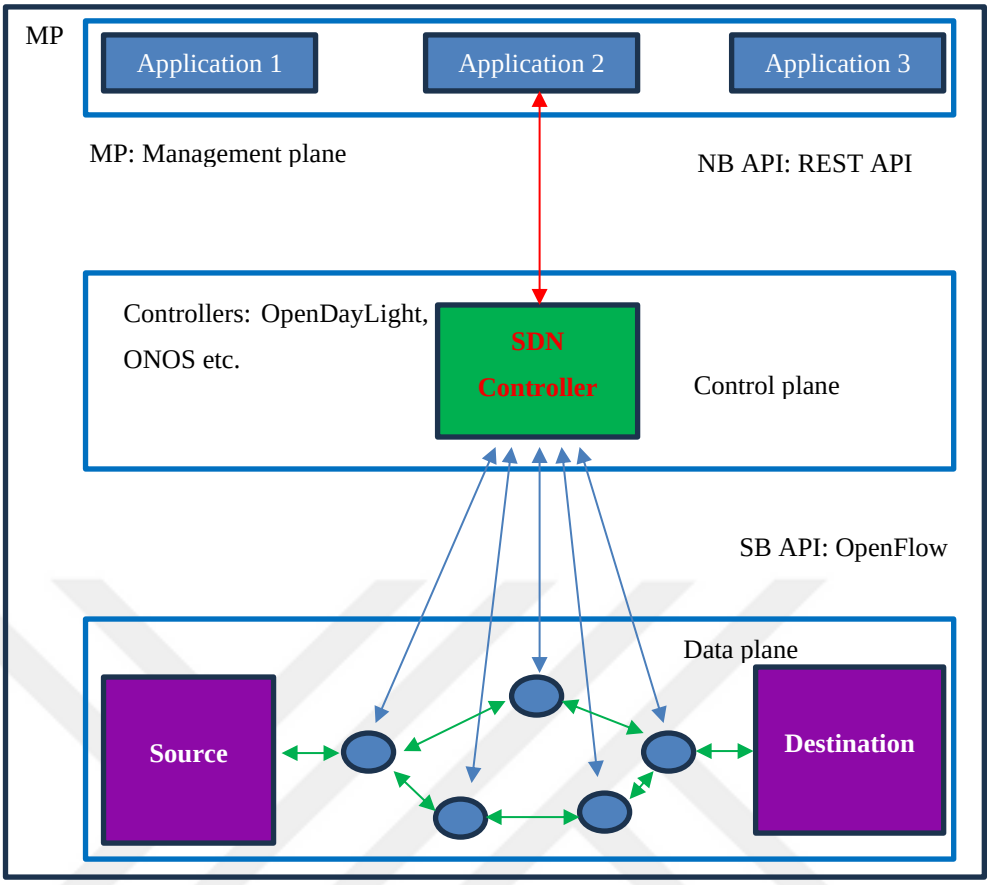


Figure 1.1. SDN layers [8].

Network operator system and the applications are both part of the system controller in a software-defined network [7], [9]. There are three: an internet portion, a control portion, and an application section. For sending and directing data, it has a three-layer design. This architecture's lowest part is concerned for data, the midst part for controlling data, finally the upper part for the applications. As can be seen, there is two-way communication between the applications part and SON part for control the flowing data. The data are located in data part, and the SON control part defines rules to transferring data [7], [9].

We aim to create an all-inclusive protection system against DDoS assaults in fluctuating network states, enabling it to adapt to recently sprouted attack channels. In this study, we evaluate the feasibility and performance of the recommended method that can protect the SDN infrastructure optimization methods through simulation-based research and then experimental analysis.

Most of the “IDS” in “IoT” must contend with challenges such as high error rates and a lengthy detection time for attacks. Using distributed platforms, such as switches based on “SDN” architecture, for the purpose of traffic analysis and attack detection is an approach that is suitable for attack detection. Another challenge of the existing methods is that learning is performed on all network traffic features, and now if the important features of the traffic are extracted and learned on them with intelligent methods, then the accuracy of the attack detection system will increase. This work displays an intelligent SBN intrusion detection system (IDS) utilizing a metaheuristic way. (BMNABC) algorithm has utilized through the controllers in the proposed system to select the features. Also, the proposed way utilizes the (ANN) to assort data into attack or non-attack classes. ML algorithms like DT, SVM, K-NN, ensembles, and NB, have been evaluated and utilized to comparing the finding. But the ANN has been obtained the greatest results.

1.1. PROBLEM STATEMENT

In modern network architectures, Software-Defined Networking (SDN) represents a significant shift by centralizing control functions and effectively separating the control plane from the data plane. Although network management and optimization gain from this centralization, there are also noticeable vulnerabilities introduced, particularly with regard to Distributed Denial-of-Service (DDoS) assaults. Such attacks can overload all network resources so that they cannot be used anymore, disrupting services entirely, and having dire consequences in situations where they are relied upon.

The growing number of Internet of Things (IoT) apparatuses leads to higher attack surfaces, so there is a need to adopt better security measures against this assault. The primary purpose of this study is to develop an intrusion detection system (IDS) tailored for SDN frameworks to address such weaknesses.

1.2. OBJECTIVES OF THE THESIS

The research’s goals are as follows:

- To develop robust models for detecting and classifying Distributed Denial-of-Service (DDoS) attacks in large-scale software-defined networks (SDNs) using advanced artificial intelligence (AI) techniques.
- To optimize network resource allocation by utilizing artificial neural networks (ANNs) and the binary multi-neighborhood artificial bee colony (BMNABC) algorithm, aiming to enhance the accuracy, reliability, and effectiveness in detecting, preventing, and mitigating DDoS attacks.

1.3.CONTRIBUTION TO LITERATURE OF THE STUDY

Global communications networks remain under constant threat from distributed denial-of-service (DDoS) attacks that deprive them of service by rendering them unavailable and degrading their intended performance [15]. DDoS attacks, protocol-based attacks, and application layer attacks are the three broad categories that can be used to classify distributed denial of service attack techniques. These techniques have been extensively documented by researchers [16].

1. Volumetric Attacks: The tendencies of these attacks are likely to congest, overloading both the target networks or services with large amounts of traffic, consuming all available bandwidth [15]. UDP floods and amplification attacks, which are very famous too, are intended to consume network resources and make it impossible for legitimate users to use the service [17].

2. Protocol-Based Attacks: These are also named state-exhaustion attacks, which utilize vulnerabilities of network protocols. For instance, SYN floods manipulate the TCP handshake process to deplete connection state tables on servers, routers, and other network devices, resulting in service disruption [18].

2. Attacks on the Application Layer: These sophisticated attacks are directed at particular applications or services by bombarding them with a large number of requests that appear to be legitimate. This approach exhausts the server's resources, such as CPU and memory [19]. Examples include HTTP floods and slowloris attacks [20].

It is still difficult to effectively mitigate large-scale distributed denial of service attacks, despite the progress that has been made in traditional security measures such as firewalls and intrusion detection systems (IDS) [21]. Such measures often prove inadequate in dynamic and programmable network environments, such as Software-Defined Networking (SDN), where their limitations are particularly pronounced [22]. Important findings from this study include:

- Creating a decentralized system for detecting intrusions in SDN architecture.
- Using the BMNABC method to enhance attack detection precision and optimize feature selection.
- Integrating deep learning methods with the BMNABC algorithm to enhance the protection of IoT services.
- In the process of conducting a comparative analysis of different machine learning techniques, demonstrating the superior performance of the ANN-based approach that has been brought forward.
- Achieving high detection accuracy, with the ANN model surpassing traditional methods in terms of accuracy, precision, recall, and F1 score.

The research contribution:

- The “DDoS” attacks in “SDN” architecture
- Detection of “DDoS” attacks with “BMNABC” algorithm.
- Training a transfer learning model in “SDN” controllers and sending the trained model to SDN switches.

Advantages:

- Ability to process large volumes of traffic for intrusion detection.
- Using SDN distributed architecture in detecting attacks.
- Detection of attacks based on important characteristics of network traffic.
- The ability to exchange information between SDN switches to detect attacks.

Disadvantages:

- Training time to create transfer model.
- The system has time overhead in the selection phase.

1.4. THESIS OUTLINE

The thesis is systematically structured into several chapters, each focusing on different facets of the research:

- Introduction: Presents an overview of the research problem, its motivations, and the objectives of the study.
- Literature Review: Explores existing research in the areas of SDN security, DDoS attack mitigation, and intrusion detection systems.
- Methodology: Outlines the proposed BMNABC algorithm, the integration with ANN, and the overall system architecture.
- Implementation: Describes the procedures that must be followed in order to put the proposed system into operation, such as the gathering of data, the extraction of features, and the training of the model.
- Evaluation: The experimental setup, as well as performance measures, are described, and a comparative analysis of the suggested system in comparison to alternative machine learning methods is presented.
- Results and Discussion: Examine the findings and highlight how successful and efficient the suggested strategy was.
- Conclusion and Upcoming Research: Provide an overview of the main conclusions, talk about the ramifications, and make recommendations for future studies.

This well-organized framework guarantees a comprehensive analysis of the issue, suggested fixes, and their real-world ramifications, providing a transparent road map for boosting SDN architecture security against DDoS assaults.

PART 2

LITERATURE REVIEW

Nowadays the paradigm shifts for the communication and information we can say are advancing with the intention of altering the established patterns of the communication networks. IoT considers as a network communication innovation that links intelligent devices. A vast network of intelligent devices and sensors surrounds us, known as the IOT. The (IoT) consists of various heterogeneous intelligent objects, each of which contains built-in smart sensors. These sophisticated, networked gadgets talk to one another without the need for human contact [24].

The Internet of Things (IoT) utilized in a wide range of industries, even in the agriculture, intelligent homes, transportation, healthcare, and education [25]. The IoT has many benefits, but it also has several drawbacks as a communication network. Security concerns are one of the obstacles facing the IoT. Due to the limited computing power of IoT network-connected objects, comprehensive security mechanisms cannot be applied to them. The inadequate security of IoT nodes and objects has made them vulnerable to various network assaults [26], [27]. For instance, IoT cameras have lax security and occasionally, owners are unable to alter the default password. Botnets are created when drones or other intelligent things become infected with various forms of malware and viruses. Machines or nodes that have been infected by malware are known as botnets, and they collaborate with other botnets to launch attacks against network services. Attacks like botnet have the goal of shutting down a server as a result of a deluge of erroneous requests [28].

An illustration of a botnet attack on the network is (DoS) attack [29]. According to studies, the global IoT market has estimated to be worth $389 billion in 2020. By 2030, it is expected that the value of the IoT network would exceed $1 trillion. It has been calculated that the Internet generates billions of bytes of data per second. Because of

its great value and volume of generated traffic, a network is an attractive target for all types of assaults, hence it is essential to create sophisticated security measures for it [30]. The network's security concerns have grown as there are now billions of IoT devices. According to projections, over 70 billion things will be connected to the IoT network in the upcoming years [31].

The linked objects can quickly get infected, acting as a botnet and a node assaulting the network. The IoT presents a wide range of complex security challenges. Operating system-level attacks include spoofing, low-level Sybil, and vulnerable physical interfaces. In the network plane, there are recurrent attacks, sinkhole attacks, and wormhole attacks. Attackers take advantage of this vulnerability and perform (DDoS) attacks to stop Internet network by overloading the target's processing power with Internet traffic [32]. IoT assaults can often quite harmful. As an illustration, the Mirai virus has infected more than 65,000 IoT devices in the first 20 hours following its release on August 1, 2016, and more than 300,000 systems were infected with malware [30], [33].

A DDoS assault using the Mirai protocol with 620 Gbps of bandwidth was launched against the website of the security consulting company Brian Krebs in September 2016. A DDoS assault with a peak bandwidth of 1.1 terabits per second that was directed at a French cloud service provider is another illustration. A Mirai version disrupt hundreds of websites, such as Netflix, Twitter, Reddit, GitHub and Reddit GitHub, for many hours in October 2016 [22],[34]. In 2020, there were over 31 billion IoT devices, yet half of them were vulnerable to serious assaults, as stated in reports from Threat Post and Security Today [22]. All of the aforementioned instances demonstrate how the IoT network is susceptible to different attacks and how serious harm is done to the network architecture. It is crucial and required to give a (NIDS) [35].

To the network to identify attacks in the IoT network. The software-based network is a packet switching system made up of controllers, several switches, and a controller. A data and control panel, as well as a number of switches, are attached to each controller. Switches for SDN have been employed as intrusion detection systems in

various research. Among the learning techniques for intrusion detection in SBN are SVM [36], DT [37], neural networks [38], deep learning [39], and neural networks [38]. In SDN switches, these studies examine network traffic and look for assaults. They discussed a case study involving network intrusion detection systems in software-based network architecture employing the idea of machine learning in the study [40].

Their study's findings demonstrate that DL techniques, a subset of ML techniques, are better able to recognize attacks and utilize them in NIDS. As stated in study [41], 2022, they presented a technique to anticipate attack patterns in the IoT network and through ML in the SON. F or find potentially harmful links and attack targets in this study, machine learning algorithms are taught using standard network threat intelligence data. The findings of the trials demonstrate that the Bayesian network algorithm is superior to DT method to identifying the threats, with an accuracy of 92.87% in detecting attacks. IIDS against cyber risks in the IoT was suggested and implemented in the study [30], 2022. A lightweight convolutional neural network model to categorize various cyber threats is the central component of their proposed intrusion detection system. The studies revealed that their suggested method consistently detects nine different sorts of cyber threats, including back doors, shell code and worms, with an average accuracy of 97.22%.[42].

This version of ID method that specific for systems control which utilizes industrial was proposed and introduced in the study [42], 2022. In [42], NIDS was designed using a CNN which reconstructed. Evaluations reveal which their suggested solution has a 97% accuracy rate for identifying network intrusion. Deep neural network-based real-time ID was suggested by researchers in the paper [43], 2022. The suggested system depends on DNN trained on 28 characteristics from the NSL-KDD dataset. The proposed method's corresponding indicators for network permeation sensitivity, specificity and accuracy are 96%, 70%, and 81%. The ability of this intrusion detection system to recognize contemporary network intrusions is a key benefit [44].

A network intrusion detection system with cuckoo search optimization was suggested in the paper [44], 2022. Experiments revealed that their suggested solution, which

combines an artificial bee colony and fuzzy clustering artificial neural network to identify intrusions, high accuracy. DL technique in the IoT was modelled and suggested in the study [45], 2022, for network intrusion detection. The suggested approach uses a SNN model to further classify the data after first identifying the most important attributes using the spider monkey optimization feature selection methodology [46].

The proposed model's accuracy for random forest classification is 94.69%. A hybrid DL pattern to recognize botnet assaults on IoT contexts was introduced and published in the paper [46], 2022. An accuracy of roughly 88% is possible for the suggested system of attack detection results. Their analysis reveals that their security and intelligence system, which makes use of a developed DL approach, has succeeded in identifying attacks which affected camera appliances that were connected to IoT applications. via choosing the best feature using feature learning via convolutional learning networks and deep learning based on LSTM networks, they offered a security and IDS to cope with DDoS threats in the study [47], 2022. This method is more accurate than IDS like ML, according to testing on proposed intrusion detection system on a number of criterion data sets and outcomes comparison in conventional models.

Phishing assaults feature a multi-stage cycle, as shown in Figure 2.1. The phisher builds a comparable phony website and impersonates the genuine website in the first phase [48, 50]. The following stage involves a phisher uploading the phony website on the Internet. Through communication medium, the hacker directs users or victims to phony webpages by sending links to them. Users access the bogus website by clicking the link, and the information they input is stolen by the fraudster [48].

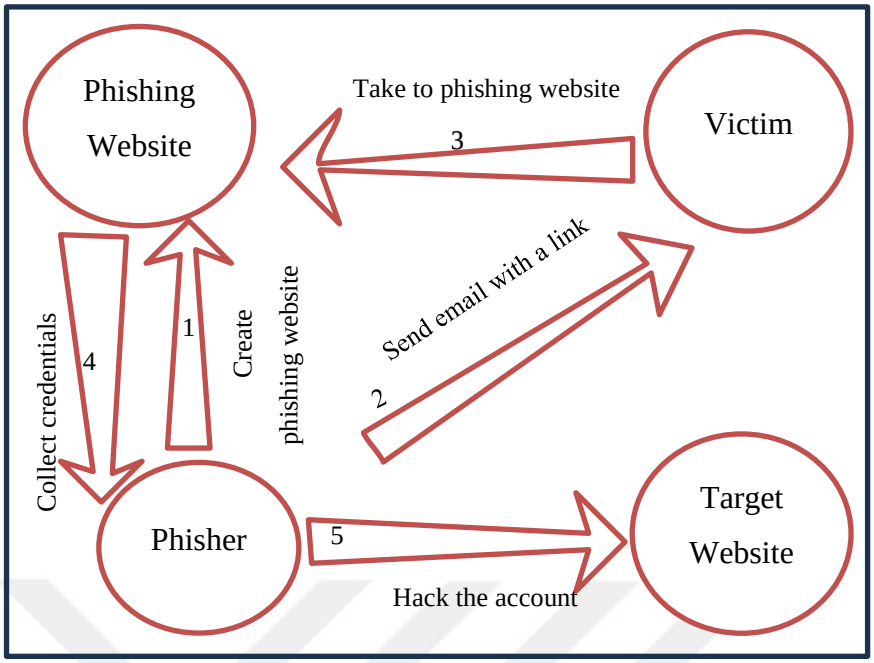


Figure 2. 1. The sequence of cyberattacks [48], [50].

Figure 2.2 shows the many methods used to identify phishing attempts. These techniques include machine learning, list-based, and similarity-based strategies.

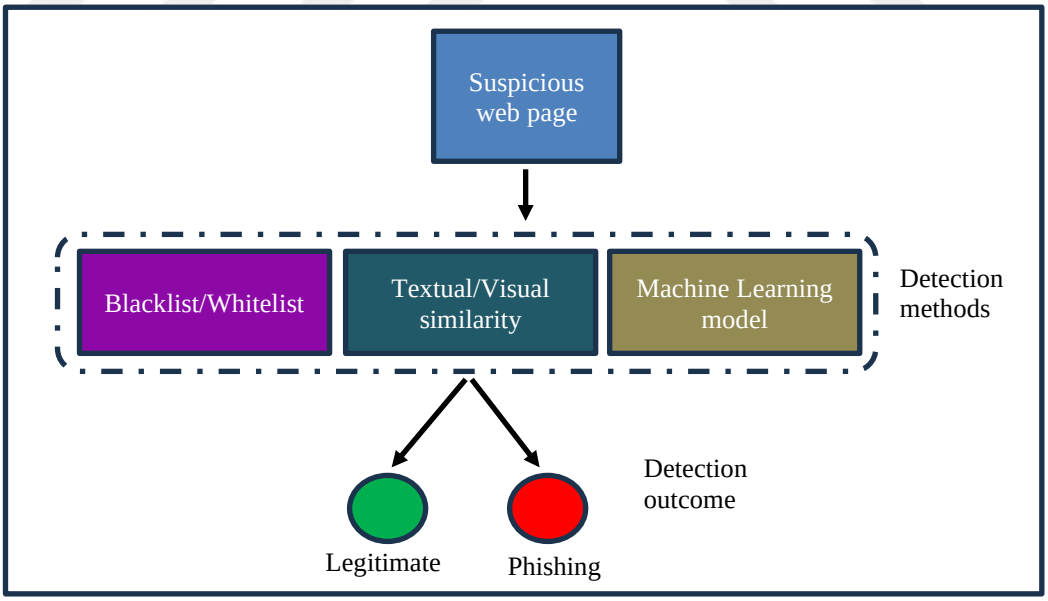


Figure 2.2. Approaches to combating phishing attacks [51].

While DL and ML offer a useful alternative by adapting to novel kinds without relying on signatures, conventional signature-based intrusion detection systems are unable to identify zero-day assaults [52], [53].

Benefits of DL and ML for Zero-Day Attack Detection [52], [53]:

- Deep learning algorithms are particularly good at identifying complex patterns that are necessary for spotting zero-day assaults that might exhibit peculiar or subtle behaviors.
- Machine learning models are able to recognize new threats and changing attack patterns since they are able to learn and adapt to new data on a continual basis.
- Deep learning models enhance overall detection efficacy by eliminating the need for manual feature engineering by automatically extracting pertinent features from raw data.

Handling False Positives and Overfitting [54]:

- Deep learning and machine learning techniques have limitations despite their potential, such as the overfitting issue that results in a high number of false positives

Techniques for mitigating include:

- Adding more variety and volume to the training set helps the model generalize more effectively to fresh data [55].
- By penalizing complex models and preventing unnecessary memory, techniques like dropout and early stopping stop overfitting [56].
- An ensemble of models minimizes the chance of overfitting while increasing performance [57].
- Post-processing methods like anomaly detection and reputation-based filtering can reduce false positives. Anomaly detection finds anomalies in predicted patterns, while reputation-based filtering takes the reliability of the data source into account [58].

Integrating Conventional and Deep Learning Methods [59]–[61], Conventional security measures are complemented by deep learning and machine learning, not replaced by them. A mixed strategy provides:

- Use the advantages of each strategy to overcome each one's shortcomings.
- Mixing various detection techniques to improve overall detection accuracy and decrease false positives.
- By adding new rules and signatures to conventional components and continuously learning from fresh data in machine learning models, the hybrid system adjusts to evolving threats.

Phishing detection methods in [62] are predicated on the analysis of human behavior and artificial intelligence. The automatic phishing detection technologies that are available on websites that mix artificial intelligence and human behavior are the subject of this study.

In [3], machine learning is applied to detect phishing websites. To detect phishing attempts, this study uses gradient boosting classifier (GBC), random forest, and decision tree techniques in combination with three feature selection algorithms. They use a dataset that contains 89 different features. The gradient-boosting classifier method has been shown to perform better than decision trees and random forests when it comes to the accuracy of attack detection, as demonstrated by experiments.

To detect phishing websites, they introduced an attribute selection technique in [63], that makes use of the particle swarm optimization (PSO) algorithm. Based on experimental results, machine learning models' ability to identify phishing assaults is improved when the PSO-based feature picking model is used. The results of the experiments show that the neural network has the best accuracy 97.81% in identifying phishing assaults.

They introduced an LSTM-based technique for email phishing detection in [5]. They have put forth a paradigm for phishing email detection that blends LSTM and federated

learning. According to the findings, their approach can produce predictions with an accuracy of 83%.

They suggest using a collection of DL types and cloud computing to detect phishing assaults in [64]. For URL analysis, they use the LSTM model; for logo analyze, they use the YOLOv2 model; and for visual similarity analyze, they applied the three-network model. The findings demonstrate that combining models yields better detection of phishing assaults than using separate models.

In order to identify phishing attacks, the author of [65] employs deep learning techniques in conjunction with hyperparameter optimisation. In order to achieve a high level of precision when identifying malware websites, the purpose of this study is to improve deductive reasoning (DL) approaches and meta-parameters. As part of this research, three different deep learning algorithm architectures—namely, CNNs, fully linked deep neural networks, and short-term memory-based recognition models—are investigated. From the results of the tests, it was determined that the model that they proposed had the highest accuracy, which was approximately 97.7%. Using the network search algorithm and the original algorithm, respectively, raised the accuracy of the models by approximately 0.1% and 1%, respectively, according to the testing.

They described a method in [66], for employing visual methods to identify phishing pages on websites pertaining to health. To identify phishing assaults, this work employs three classifiers: DT, a random forest, and a SVM. The DT achieves the highest precision in identifying phishing assaults, according to experiments.

CNN-based method for determine phishing assaults on social media platforms is presented in [67]. Phishing via simple notification services (SNS) is one example of a social engineering assault that preys on people's emotions and confidence. The perpetrator of these attacks develops a strong emotional bond with the victims. CNNs are used in this study as the Telegram chatbot's moniker. Investigations showed that when it comes to phishing attack detection, the Text-CNN approach outperforms LSTM in terms of accuracy.

A deep learning technique for spotting phishing websites based on visual resemblance can be found in [68]. The extraction of visual cues is a significant difficulty in phishing detection systems that rely on visual similarity. In order to extract features for machine learning algorithms, this work employs a transfer learning technique. The experimental results show that accuracy findings in phishing attack detection are obtained by combining VGG16 with an algorithm based on machine learning.

Temporal convolutional network (TCN) can be used for phishing URL detection [69]. To detect phishing URLs, they employ a novel deep learning method called TCN with word embedding. According to experimental results, their approach has an accuracy and sensitivity of 98.95% and 98%, respectively, in detecting phishing IP addresses.
A Bi-LSTM neural network is suggested in [70] for phishing attack detection of spam and bogus emails sundry ML and DL models are utilized in this research., such as the SVM, long-term memory, short-term memory, bidirectional LSTM, random forest classifier, and "ANN". Assessments show that deep learning approaches are more accurate than machine learning approaches.

A method for creating fake URLs utilizing a GAN is presented in [71] in order to identify phishing URLs. Current URL databases need to be balanced because they contain few samples. The researchers proposed training a GAN network, called WGAN-GP, to generate malicious URLs from pre-existing data on phishing URLs in order to solve this problem. Phishing attacks were detected using LSTM and GRU classifiers.

DL-based approach for determine zero-day phishing assaults may be found in [72]. In order to identify phishing assaults, they presented a system in this study that combines deep learning with logically coded domain knowledge. They suggested using the standard learning approach in conjunction with logical and neural classifiers. According to the testing, their approach raises the sensitivity index by roughly 3%.

The study described in [73] stands out when it comes to the analysis of URL analysis. In order to identify potentially dangerous URLs, the researchers methodically developed a robust ensemble learning model by applying a multivariate filter-based

feature selection technique. Their method, which combined statistical t-tests with correlation feature selection to identify critical features, produced noteworthy results. The accuracy rates that were attained were astounding; they were 97% in the first dataset and a staggering 99.25% in the second.

They suggest using a DL algorithm and natural language processing to identify phishing websites in [74]. The goal of the proposed study is to create an autonomous and hybrid model that can detect and categorize phishing in URL and web page content using a CNN algorithm in deep learning and a random forest method in ML

Their method outperforms conventional ML techniques in terms of phishing attack detection accuracy. The examined studies in phishing detection are compared in Table 2.1. The majority of studies employ ML and DL techniques to attack detection, according to an examination of related studies and publications. The limitations of visual approaches for phishing attack detection are numerous, and the analysis of website images by visual algorithms is time-consuming. The accurate of ML techniques is mediocre if attribute selection is absent. Techniques for balancing data sets are crucial for improving the precision of DL and ML algorithms. The majority of research focuses on URL addresses, but the properties of content, domains, search engines, and source code of online pages all include useful information for phishing attack detection. To address these issues, the suggested solution offers a DL-based strategy which collects swarm intelligence and data set balance. Analyzing related works to identify phishing attempts reveals the following difficulties:

- When deep learning techniques are taught on uneven data, their accuracy is reduced.
- Training deep learning algorithms that do not reduce the dimensionality of the input takes a lot of time.
- The accuracy of identifying phishing assaults is decreased when deep learning techniques lacking in optimization are employed.
- Simple GA and PSO algorithms are used in many research to identify phishing attacks, although they are not very good at s searching the issue space. Compared to earlier metaheuristic algorithms like GA and PSO, swarm

intelligence algorithms that have been presented recently exhibit greater complexity and robust modeling.

Table 2.1. Strategies, benefits, and drawbacks of works that are linked to this one

Ref	Existing Approaches	Disadvantages	Advantages	Methodology
[32]	Static controller placement lacks adaptability	Complex to implement and maintain.	Achieves 99.99% accuracy in controller placement. Provides rapid response to DDoS attacks.	Use real-time monitoring and machine learning to detect and respond to threats
[33]	Traditional DDoS protectors often discard normal traffic	Requires precise priority control of network frames. Implementation complexity with multiple vendor integration	Prevents discarding normal traffic within seconds. Blocks attack traffic in milliseconds	Develop a DDoS suppression system that prioritizes network frames to reduce discarding normal IoT traffic
[75]	Demands additional data collecting and analysis to be carried out.	An investigation of the actions of the phisher has not been carried out.	Using human behavior	A comparison of human behavior and artificial intelligence
[76]	Could call for additional adjustments to the parameters	Unbalanced data set	Accuracy is about 97.81%	PSO-based feature selection
[77]	Requires significant infrastructure and expertise	High complexity	Both the content and the visual appearance are analyzed.	Computing in the cloud through the combination of deep learning models
[65]	Could necessitate additional time and resources needed for optimization	high overhead	Higher in precision than both CNN and LSTM	The optimization of deep learning hyperparameters using deep learning and deep learning
[66]	May produce results that are less easily interpretable	Time complexity	With the help of the decision tree, appropriate precision	Learning with machine learning and visual approaches

[67]	Could call for additional data for training purposes	Unbalanced	LSTM is more accurate than it is.	Text-CNN
[69]	It's possible that more data is needed for good training.	Numerous hours spent in the gym	%98.95accuracy	A neural network with temporal convolutional layers
[78]	This could result in a greater computing cost.	Unbalanced data set	LSTM is more accurate than it is.	Bi-LSTM
[79]	Could necessitate a more involved data preprocessing procedure	Absence of utilization of content aspects	Balancing the dataset	GAN network intended for the detection of phishing URLs
[80]	Could necessitate additional time and resources needed for optimization	a lack of feature selection as well as a lack of an equal distribution of the data collection	Enhancing the sensitivity index by around three percent	Learning at a deep level and knowledge of programming
[81]	Have the potential to have a greater computational cost	Lack of content analysis	High accuracy	Sustainable group learning
[82]	This could result in a greater computing cost.	No feature selection	High accuracy	Natural language processing and methods
[83]	Traditional network security methods are less effective for SDN	Focuses primarily on new-flow based DDoS attacks	Improves understanding of SDN vulnerabilities	Classify security vulnerabilities in SDN exposed by new-flow based DDoS attacks
[84]	Statistical methods analyze traffic patterns	May not cover all possible DDoS attack vectors	Comprehensive review of DDoS strategies in SDN	Conduct a systematic survey of DDoS detection and mitigation strategies in SDN. Provide a taxonomy of detection strategies

				and emerging approaches

As a whole, the body of research emphasises the growing significance of developing efficient defence mechanisms against distributed denial of service attacks (DDoS) that are adapted to the specific characteristics of software-defined networking (SDN) architectures. For the purpose of enhancing the resilience of "SDN" infrastructures against "DDoS" attacks and mitigating the impact that these attacks have on network performance and availability, researchers intend to integrate artificial intelligence techniques such as "ANNs" with optimisation algorithms such as "BMNABC." On the other hand, the research needs to validate the effectiveness of these approaches in real-world deployments of "SDN" and address emerging challenges in the mitigation of "DDoS."

2.1. SOFTWARE-DEFINED NETWORKING

The approach to network management that allows the network to be dynamically and efficiently configured programmatically is referred to as software-defined networking, or SDN for short. More comparable to cloud computing than traditional network management, this technology is used to improve network performance and monitor its health [85]. The term “SDN” also refers to a technology that aims to improve the static architecture of traditional networks. Network intelligence can also be integrated into a single network component by using this network component. The process of forwarding network packets, known as the data plane, is separated from the process of routing network packets, known as the control plane 2. This allows the task to be completed successfully [86].

It is the control plane, which may consist of one or more controllers, that contains all of the intelligence that is necessary. When it comes to the "SDN" network, these controllers are often referred to as the "brains." On the other hand, centralisation is associated with a number of drawbacks that are associated with security, scalability, and other aspects of elasticity [85][87].

“SDNs” is a new generation of network architecture that seeks to make the administration of the network easier by providing centralization [82]. Unlike the traditional network, which shares intelligence between many devices, this SDN has separated the control plane from the data plane [82]. Having said that such architectural differences will increase adaptability, scalability, and operational efficiency by allowing network administrators to use software-based controllers that permit programmatic regulation of network behavior in a centralized way [88].

Since OpenFlow was first presented in 2011, the term "SDN" has been routinely associated with the OpenFlow protocol. This protocol enables remote communication with network plane elements in order to determine the path that network packets take across network switches. OpenFlow was initially introduced in 2011. The term, on the other hand, has also been utilised by proprietary systems continuously since the year 2012 [89] [90] Platforms such as the Open Network Environment from Cisco Systems and the network virtualisation platform from Nicira are two examples of types of platforms that fall into this category. SD-WAN is a technology that is very similar to other technologies when it is applied to a wide area network (WAN) [91].Table 2.2 Summary of core SDN components with description and links to further reading.

Table 2.2. Summarizing the key components of Software-Defined Networks (SDNs)

Ref	Key Component	Description
[92]	Controller	Orchestrates network traffic flow and routes data packets throughout the network. Interfaces with network devices via southbound APIs.
[93]	Data Plane	Consists of network switches and routers that are accountable for the forwarding of packets and are directed by the software-defined networking controller.
[94]	Southbound APIs	Interfaces that allow for transmission of data between the SDN controller and the various network devices. Transmit control commands and collect network status information.
[88]	Northbound APIs	Expose SDN capabilities to applications and network services, facilitating integration with higher-level applications, orchestration systems, and management tools.
[95]	Applications and Services	Provide support for a wide range of capabilities, including traffic engineering, security services, load balancing, and utilities for network monitoring.

[96]	Virtualization	Enables multiple virtual networks over a shared physical infrastructure, enhancing resource efficiency and enabling tailored network services.
[97]	OpenFlow Protocol	To facilitate centralised management of network traffic flows, a standard for communication between software-defined networking controllers and network devices has been developed.

While traditional models of networking bring certain benefits, SDNs introduce a lot more, such as improved network agility, security through centralized policy enforcement, easy provisioning and administration of the network, and support for dynamic and scalable network architectures. SDNs are central to the achievement of shifting requirements in many industries and applications due to rising demand for ever-more responsive and flexible networks [98].

2.2. DEFINITION OF DOS AND DDOS

They are two similar attackers aims to overwhelm the systems by huge traffic data and exploit the systems vulnerability.

- **Definition of DoS**

The term "denial of service" (DoS) refers to a type of cyberattack in which the attacker attempts to prevent a machine or network resource from being accessible to the users for whom it was designed. This is accomplished by temporarily or permanently disrupting the services of a host that is connected to the Internet. A distributed denial of service attack's primary goal is to make a service or system inaccessible to the people who are supposed to use it. This can frequently result in inconvenience, financial loss, or damage to a company's reputation. An Overabundance of Resources As a result of the attacker flooding the target with an excessive number of requests, the target's capacity to process legitimate requests is overwhelmed. It is possible for this to involve the consumption of bandwidth, server resources, or other essential components of the system.Taking Advantage of Weaknesses Attackers will sometimes take advantage of particular vulnerabilities in the system they are targeting in order to bring about the system's failure or instability [99], [100].

- **Definition of DDoS**

Distributed denial of service, also known as DDoS refers to a malicious attempt by an attacker to prevent legitimate users from accessing a server or network resource by flooding it with artificial traffic. This is done in an attempt to prevent legitimate users from using the server or resource [101].

DDoS, is the result of a cyber-attack in which a server or network resource is rendered inaccessible to traffic that is expected to come from legitimate users. Denial of service is a consequence of an attack, which is defined as the intentional disruption of a target host that is connected to the internet by an individual who is the perpetrator of the attack (the attacker) [101].

2.3. OVERVIEW OF DDOS ATTACKS

“DDoS” attacks still represent a serious risk to global network infrastructures. The development of “DDoS” assault methods, such as volumetric, protocol-based, and application-layer attacks, has been the subject of numerous studies. Traditional means of protecting against large-scale “DDoS” attacks, such as firewalls and intrusion detection systems, have failed mainly because these measures do not respond effectively to in dynamic and programmable network environments like “SDN” [102].

2.3.1. Challenges in SDN Security

“DDoS" attackers regard the centralized control plane of “SDN” as a tempting target since it has additional vulnerabilities [103]. There are several challenges that are involved in defending against Distributed Denial of Service attacks on “SDN” infrastructure, some of which include the limiting visibility into network traffic beside any single point of failure exploitation potential used for identifying and mitigating threats, limited visibility into network traffic besides any single point of failure exploitation potential [103].

2.4. ARTIFICIAL NEURAL NETWORKS (ANNs)

“ANNs” are advanced computational frameworks inspired by the human brain's neural structure. They have the ability to process data and learn how to differentiate between various patterns since they are made up of interconnected nodes or neurones. These networks form the basis of Artificial Intelligence and have found a deep root in changing many disciplines, from computer vision to natural language processing to predictive analytics [75]–[77].

2.4.1. Structure of ANNs

There are typically three distinct sorts of layers that make up an “ANN” [18].

- Input Layer: This layer is responsible for receiving the data that is input. Each neurone in this layer relates to a different characteristic of the data that is being input.
- Hidden Layers: These layers are located between the input and output layers, and it is in these layers that sophisticated computations are carried out in order to extract features from the data that is provided as input. At this point, the architecture of the network can be defined by the number of hidden layers and neurones, as well as the configuration of those neurones.
- Output Layer: This layer of the network is the final one, and it is responsible for producing the final output of the computer network. It is the number of variables that are being predicted that determines the number of neurones that are contained within this layer.

2.4.2. Functioning of ANNs

In order to train “ANNs”, the weights of the interconnections between the neurones are adjusted. Backpropagation is the most widely used method for this training process. It modifies these weights in order to narrow the gap between the output of a network and the output that is desired [78], [79].

Artificial Neural Networks are very powerful tools with numerous applications, capable of learning from data and modeling complex relationships. Their intrinsic difficulties notwithstanding, ongoing research and improvements in computational power continue to raise their potential, as well as expand their applicability [80], [81].

2.4.3. Artificial Intelligence for DDoS Mitigation

In particular, “ANNs”have shown promise in enhancing “DDoS” detection capacities. “ANNs” enable accurate detection of abnormalities that may indicate “DDoS” attacks and enable real-time monitoring of network traffic patterns. While several research have examined the usage of “ANNs” in traditional network situations, their effectiveness in “SDN” systems remains unknown [80], [81].

2.5. BINARY MULTI-NEIGHBORHOOD ARTIFICIAL BEE COLONY BINARY MULTI-NEIGHBORHOOD ARTIFICIAL BEE COLONY

The Multi-Neighborhood Composed of Binary In order to solve binary optimisation issues, the Artificial Bee Colony algorithm is a modified version of the original Artificial Bee Colony algorithm. In this respect, the “BMN-ABC” enriches the simple ABC by involving multiple neighborhoods and provides an overall enhancement in search speed and convergence performance in binary space [104].

In the case of the classical “ABC” algorithm, an artificial population of bees performs the foraging operation of honey bees to find the best possible solutions. This process goes through three stages: scout bees, observer bees, and employed bees, one by one. In their community, each individual bee explores the solution space, evaluates the possible solutions, and communicates with other bees to guide the entire search process [104].

“BMN-ABC” encodes the solutions as binary strings. The main novelty of “BMN-ABC” is the use of more than one neighborhood, which allows for a better exploration of the solution space. Then each neighborhood is applied with a different procedure to flip bits in binary strings to come up with new answers. These diversified regions are

utilized by the “BMN-ABC” algorithm in dynamically adjusting the search strategy, preventing premature convergence and enhancing optimization efficiency through balancing exploration-exploitation [104].

A multi-neighborhood binary system The contribution that Artificial Bee Colony makes to binary optimisation is among the most helpful. Multiple neighborhood structures have been integrated with adaptive selection into the “BMN-ABC” algorithm to improve search and convergence speed over the traditional “ABC”. This enhances its power as a tool that is robust in solving complicated binary optimization problems with better performance and flexibility than standard approaches. To that respect, the “BMN-ABC” algorithm can be utilized for solving different binary optimization problems by researchers and practitioners due to the robust and efficient optimization capability of the approach [104]–[106].

2.5.1. Performance Metrics

One should have some metrics to quantify how good the machine learning model is, especially for classification tasks. These measures are helpful in gaining an insight of how accurately the model is predicting the future [107].

Let's discuss some of the most common performance metrics:

2.5.1.1. Confusion Matrix

Before exploring specific metrics, it's important to grasp the concept of the confusion matrix [118]. The confusion matrix representation is shown in Table 2.3.

Table 2.3. Confusion matrix representation [108]

	Predicted Positive	**Predicted Negative**	
Positive in actuality	“TP” Ture Positive	“FN” False Negative	Sensitivity $\frac{TP}{TP + FN}$
Negative in actuality	“FP” False Positive	“TN” True Negative	Specificity

			$\frac{TN}{TN + FP}$
	Precision $\frac{TP}{TP + FN}$	Negative Predictive value	Accuracy $\frac{TP + TN}{TP + TN + FP + FN}$

2.5.1.2. Accuracy

Accuracy is the most straightforward metric, as it is calculated by taking into account both the total number of predictions and the number of predictions that were correct [109]–[112]. Despite the fact that accuracy provides a glimpse, it can be misleading because it points to problems that are associated with imbalanced datasets in which one class is significantly more numerous than the other. An example of a model that would demonstrate a high level of accuracy would be one that identifies all instances as belonging to the majority class. However, if this model were to be applied to the minority class, it would be unsuccessful [109], [110].

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.1)$$

2.5.1.3. Precision

Precision can be defined as the number of true positives that are based on all of the positive predictions that were made. In the event that the precision is high, one can be certain that the model will be accurate when it makes a prediction about the positive case happening. In regions where the cost of false positives is relatively high, this measure is especially important [109], [110].

$$Pr\,e\,cision = \frac{TP}{TP+FP} \quad (2.2)$$

2.5.1.4. Recall (Sensitivity or True Positive Rate)

The proportion of actual positive cases that were predicted by the model out of the total number of positive cases is referred to as the recall. In light of this, the

effectiveness of the model in identifying all of the positive cases is evaluated by this methodology. This measure is helpful in situations where the cost of missing a positive case is high, so it is similar to the majority of medical diagnosis studies in that it is helpful in those situations [115]–[118], [120], [122].

$$Sensitivity = \frac{TP}{TP+FN} \tag{2.3}$$

2.5.1.5. F1-Score

The F1-score is the harmonic average of both recall and precision; as a result, it makes an effort to strike a balance between the two. The use of this tool is extremely beneficial in situations where there is an imbalance between the positive and negative classes. A score of one on the F1 gives the impression of perfect precision and perfect recall, whereas a score of zero indicates either ideal imprecision or perfect recall. The F1 score comes in especially handy in cases where one wants to balance between precision and recall, so neither metric overshadows the other [115]–[118], [120], [123].

$$F1 = 2 \times \frac{Re\,call \times Pr\,ecision}{Re\,call + Pr\,ecision} \tag{2.4}$$

2.6. COMPARISON WITH OTHER AI TOOLS

Comparing the artificial neural networks with the binary multi-neighbor artificial bee colony, we will also compare five other artificial intelligence tools, which are as follows: The following are some examples of ensemble methods: K-nearest neighbours (K-NN), decision tree (DT), support vector machine (SVM), naive Bayes (NB), and similar techniques:

2.6.1. K-Nearest Neighbors (K-NN)

Strong points: “KNN” stands for "k-nearest neighbours." One of the advantages of KNN is that it is one of the most straightforward algorithms that can be utilised in the process of supervised machine learning. The “KNN” algorithm is based on the assumption that there are other objects in the vicinity that are similar. KNN has some restrictions. Image processing, defect detection, classification, clustering, grouping, colour prediction, and fibre identification are just some of the applications that regularly make use of k-nearest neighbours [123-124] In addition to this, it is a particularly straightforward non-parametric algorithm that is frequently utilised for classification and regression [124] “KNN” is a learner that is based on instances [125] The use of a small k is typically recommended because it helps to ensure that the data point in question does not contain an excessive number of other data points that could potentially obscure its true nature [126].

Weaknesses: It is characterized as a lazy learner algorithm because of its slowness [125] It also becomes slower with increasing data size [124] Furthermore, when the value of k is high, it can result in overfitting and model instability; consequently, it is necessary to select values that are suitable for a particular application [126].
Contrast: However, in the case of multiple and abundant data, “BMNABC” performs better in terms of solution accuracy and convergence speed. In addition, the “BMNABC” algorithm applies near and far neighbour information by utilising a new probability function in the first and second stages of the process. In comparison to the “KNN” algorithm, it conducts a search that is more conscious, and it achieves satisfactory results in solving problems [105] “KNN” is a method that is accurate, but it requires a lot of computational power, which makes it less suitable for large datasets. The “ANN”, on the other hand, provides a balance between accuracy and efficiency, which makes it more suitable for applications that are on a large scale [127].

2.6.2. Decision Tree (DT)

Strengths: Simple and Straightforward Decision-Making Process Trees are simple and easy to understand. Because they are based on how people actually make decisions, they are simple to understand and depict. The tree's nodes stand for feature-based decisions, and the branches for potential outcomes [128]. Neither feature scaling nor

normalisation is necessary for DT. Since the algorithm can naturally process both numerical and categorical data, they deal directly with raw data [128]. There is no need to preprocess numerical or categorical data before using a DT. Because of this, they can be used with a variety of datasets [129]. When it comes to outliers, DT are pretty resilient. Unlike methods that are sensitive to data variability, methods that split the data into subsets based on feature values mean that extreme values don't have as much of an impact on the structure of the tree [130].Understanding the significance of features can be aided by DT. Tree split analysis helps with feature selection and dimensionality reduction by revealing which features are most important for prediction [131]. When it comes to classification and regression, DT are your best bet. Complex decision-making processes and a wide range of prediction problems are within their capabilities [136]. For complicated datasets with non-linear relationships, DT are a good fit because they can capture these relationships [133]. DT can also be considered easy to see and understand. When it comes to communicating results and understanding how the model behaves, the tree structure provides a clear picture of the decision-making process [128].

Weaknesses: Overfitting is a common problem with DT, particularly with deep trees. When a model gets overly complicated and begins to generalise only to the training data rather than capturing noise in the actual data, this phenomenon is called overfitting [128]. The Decision Tree's structure can be drastically altered by making minor adjustments to the training data. Decision Trees can exhibit substantial variation across data subsets due to this instability [134]. Feature DTs with more levels or categories may have an advantage. This bias arises because it is easier for features with more levels to dominate the tree-building process, as they offer more opportunities for splitting [135]. DT might not be the best algorithm for capturing complicated data relationships, particularly if those relationships are very non-linear. They aren't very good at solving problems with complex feature interactions [133]. Without adequate management, DT can fail when faced with continuous variables. When optimising for continuous variables, it may be necessary to resort to more complex methods because splitting them can be less natural [129]. Construction Decision When dealing with massive datasets, trees, especially those that grow extremely deep, can become computationally expensive. Training times and memory usage can be affected by this

complexity [136]. When there is an imbalance in the classes in a dataset, DT can be biassed. They may do badly on the minority class because they overfit to the majority [137].

Comparison : The interpretation of DTs is typically straightforward, and they are able to process both numerical and categorical data. They are able to produce satisfactory results with structured data and offer insights into the significance of features [128, 132]. “ANNs” are able to demonstrate a high degree of flexibility and the ability to capture complex nonlinear relationships. In general, they provide superior accuracy performance for complex datasets and tasks, particularly when dealing with large amounts of data and when employing deep structures. In general, they provide superior accuracy performance for complex datasets and tasks, particularly when dealing with large amounts of data and when employing deep structures [138, 139]. For the most part, the “BM-NABC” algorithm is utilised for the purpose of feature identification and optimisation. Through the selection of pertinent features and the avoidance of overfitting, it contributes to the improvement of model performance, which in turn can improve accuracy [140, 141].

2.6.3. Support Vector Machine (SVM)

Strengths: “SVMs” are highly effective for data with many features (high-dimensional spaces) because they focus on finding the optimal hyperplane that maximizes the margin between classes [142]. SVMs include regularization parameters that help prevent overfitting by balancing margin size and classification error [143]. The use of kernel functions allows SVMs to handle non-linear data by mapping input features into higher-dimensional spaces [144].

Weaknesses: Training SVMs can be computationally expensive, especially with large datasets [145]. SVMs require careful tuning of parameters (e.g., regularization parameter C, kernel parameters), which can be challenging [146]. The training process can become impractical for very large datasets due to high computational and memory requirements [147].

Comparison : Although “SVM” is highly effective in data with multiple features, it [142]. “BM-NABC” excels in feature selection by identifying the most relevant features, which can help improve model performance and reduce overfitting [140]. It also provides optimization capabilities for feature selection, which can be useful when combined with other algorithms to improve overall performance [141]. “ANNs” excel in that they can model complex nonlinear relationships and interactions between features through multiple layers of neurons [138].They are also extremely flexible and can be adapted to different types of data and tasks by adjusting the network architecture and training parameters [139]. In addition, deep learning models, especially, perform exceptionally well with large datasets and can achieve state-of-the-art results in many domains [148].

2.6.4. Naive Bayes (NB)

Strength: Naive Bayes classifiers are computationally efficient and straightforward to implement. They work well with large datasets and provide quick predictions [149]. Effective when dealing with categorical data and can handle large feature sets with relative ease [132]. Naive Bayes tends to perform reasonably well even if some features are irrelevant, as it assumes feature independence [150].

Weakness: Naive Bayes assumes that features are independent given the class, which is often not true in complex network traffic, potentially leading to reduced accuracy [151]. May struggle with complex patterns and interactions in network traffic data where dependencies between features are significant [152].

Comparison : Neural networks can learn complex nonlinear relationships and interactions in network traffic data, resulting in high accuracy [138]. They are also very flexible and can be adapted to different types of data through different architectures, such as convolutional neural networks (CNNs) for spatial data or recurrent neural networks (RNNs) for sequential data [139].In addition, neural networks perform well with large and complex datasets, which is natural in network traffic scenarios [148]. In short, neural networks are generally more accurate and reliable for complex network traffic patterns due to their ability to learn complex data

relationships [138, 139, 148].Naive Bayes is simpler and faster but may not perform well with complex data [149].”BM-NABC” has the advantage of improving feature selection, which may improve the performance of classification algorithms used in network traffic analysis by focusing on the most relevant features [140].When used in conjunction with classifiers such as “SVMs” or “ANNs”, “BM-NABC” can improve overall classification performance by selecting relevant features [141].It enhances classification performance by improving feature selection, which may improve the accuracy of classifiers used in network traffic analysis. However, it is not an independent classifier and depends on the quality of the associated classifier [140, 141]. But NB is suitable for simpler classification tasks with categorical data but may struggle to handle complex patterns present in network traffic due to the assumption of their independence. It provides fast and efficient processing but lacks the sophistication needed for complex data interactions [132, 149, 150].

2.6.5. Ensemble Methods

Strength: Ensemble methods, such as Random Forests and Gradient Boosting Machines, aggregate the predictions from multiple models to improve overall accuracy and robustness against overfitting [134].Ensemble methods can effectively handle various data types and feature spaces, making them adaptable to diverse network traffic patterns [128]. By combining predictions from multiple models, ensemble methods reduce the risk of overfitting compared to single models [154].

Weakness: Ensemble methods can be computationally expensive and may not be as efficient for real-time applications due to the need to aggregate multiple models' predictions [155]. Ensemble models, particularly those involving many base learners, can be difficult to interpret, which might be a drawback for network security applications requiring explainable decisions [156].

Comparison: Ensemble methods: typically offer high accuracy and robustness but can be computationally demanding, making them less ideal for real-time applications in “SDN” security. While “BM-NABC” enhances feature selection and classification performance, which can improve accuracy when combined with powerful classifiers

such as artificial neural networks. However, it is not a standalone classification method and can be computationally heavy. Artificial neural networks generally provide the highest accuracy for complex network traffic patterns and adapt well to new threats. They are very efficient at real-time processing but require significant computational resources and careful management to avoid overfitting [134, 140, 149, 157]. In addition, “ANNs” are well suited for real-time applications with the right infrastructure, thanks to their ability to learn and adapt to new data. “ANNs” enhanced with BM-NABC can deliver superior performance by optimizing feature selection, making them highly effective in dynamic environments such as “SDN”. Ensemble methods, while accurate, may face challenges in real-time adaptability due to their computational requirements. “ANNs” enhanced with “BM-NABC” are a powerful combination for SDN security, providing high accuracy and adaptability to complex network traffic patterns. They leverage the strengths of “ANNs” to recognize detailed patterns and the optimization capabilities of “BM-NABC” to improve feature selection, making them a competitive option compared to traditional ensemble methods [134, 140, 149, 157].

2.6.6. Selection of BMNABC and ANNs

Optimization Efficiency: “ANNs”, especially deep learning models, are optimized through backpropagation and gradient descent, which are well-suited for learning complex patterns in large datasets [138] “BM-NABC” excels in optimization tasks, particularly in feature selection and dimensionality reduction. It uses a modified artificial bee colony algorithm to explore the feature space effectively and efficiently [140].

Feature Selection: Advanced neural networks, particularly deep learning models, can automatically learn and extract features from raw data. This reduces the need for explicit feature selection [139] “BM-NABC” is specifically designed for feature selection. It optimizes the selection of relevant features, which can improve the performance of classification models by focusing on the most significant features [140].

Adaptive Learning: “ANNs”, particularly when used in online or incremental learning modes, can adapt to new data patterns and evolving threats in real-time. This is crucial for dynamic environments like network security [162] “BM-NABC” can adapt by optimizing the selection of features based on changing data characteristics or problem requirements, improving the relevance of features over time [140].

2.6.7. Relationships of BMNABC and ANNs with other tools (limitations of BMNABC and ANNs)

“BM-NABC” is efficient in optimizing feature selection but can be computationally heavy. “ANNs” provide versatile optimization through learning algorithms but require significant computational resources.”BM-NABC” specializes in feature selection, which can significantly improve the performance of models like “ANNs” when used together.

ANNs can automatically learn features from raw data, reducing the need for explicit feature selection but may not always focus on the most relevant features.”BM-NABC” adapts through optimization but does not directly handle real-time learning from new data.ANNs are highly adaptable to new data and can continuously learn, making them suitable for dynamic environments but requiring careful management to avoid overfitting.”BM-NABC” and “ANNs” can complement each other well, with “BM-NABC “optimizing feature sets and ANNs handling complex pattern recognition and real-time learning [138, 139, 140, 159, 160, 161].

2.6.8. Compensation by Other Tools

“K-NN” and Decision Trees can be useful for preliminary analysis and feature selection. “K-NN” provides a quick understanding of class separability and feature relevance, while Decision Trees offer insights into feature importance and data structure. Also “BM-NABC” can leverage insights from these preliminary methods to perform more refined feature selection, optimizing the feature set for better performance in subsequent modeling tasks [129, 135, 140, 162,163, 165].

“SVMs” are robust in high-dimensional spaces and are effective for certain types of data, especially with clear margins of separation. They can handle complex feature spaces through the kernel trick. “ANNs” excel at learning complex patterns and can benefit from “SVMs'” capabilities in high-dimensional feature spaces. Combining “SVMs” with “ANNs” can leverage the strengths of both, improving performance across different types of tasks [142, 144, 166, 167, 168, 169,170].

Naive Bayes is indeed useful for providing quick and simple baseline models. Its efficiency and ease of use make it a valuable tool in early research stages, allowing for rapid experimentation and comparison with more complex models. Its primary advantage lies in its simplicity and speed, making it an excellent starting point for model development. However, its performance can be limited by the independence assumption, and it might not capture complex feature dependencies as effectively as more advanced methods [149, 150, 149, 150, 169, 171].

Combining “ANNs” with ensemble techniques effectively enhances model performance and mitigates overfitting. Ensemble methods, such as bagging, boosting, and stacking, leverage the strengths of multiple models to achieve better accuracy, robustness, and stability. These techniques reduce the impact of overfitting by averaging out errors and biases from individual models [134, 138, 172, 173, 174, 175, 176].

PART 3

MATERIALS AND METHODS

3.1. ARTIFICIAL NEURAL NETWORK-BASED DOS AND DDOS ATTACK DETECTION

While there has been an increase in the reliance on internet-based services, there has also been an increase in the number of cyber threats that take the form of "DoS" and "DDoS" attacks. This type of attack has resulted in severe damage not only to the user but also to the service providers, as it is designed to overwhelm the availability of service by means of traffic overload. In order to address the ever-changing nature of these threats, traditional methods of detection and mitigation frequently fall short. When it comes to detecting denial of service and distributed denial of service attacks, “ANNs” offer a promising solution due to their capacity to learn and adapt to new patterns. This makes them particularly effective.

Working Mechanism, the process of detecting “DoS” and “DDoS” attacks using “ANNs” involves several key steps:

- Data recalling: The data in the dataset is retrieved from the network traffic. This particular dataset includes numerical data that includes a variety of characteristics of network traffic records, including protocol type, service, flag, source bytes, and other similar characteristics. The dataset was obtained by clicking on the following link:
  (https://www.kaggle.com/datasets/hassan06/nslkdd)
- Feature dimension reduction: In this step the features of the dataset has been reduced by using the BMNABC method that has high efficiency in finding the best and optimum features.

- Use the machine learning and “ANN" to classifications: The “SVM”, “DT”, Ensemble, Navie Base, “KNN”, and “ANN” are some of the machine learning methods that are utilised in this stage of the process.

- Training the “ANN” and machine learning methods: The feature matrix that obtained from “BMNABC” is used to train the “ANN” and other machine learning methods. During this phase, the “ANN” learns to differentiate between normal and non-attack traffic by adjusting the weights of the connections between neurons. Typically, supervised learning is employed, with the model trained on labeled data containing examples of both normal and attack traffic.
- Detection and Classification: Once trained, the “ANN” monitors network. As new data flows in, the “ANN” processes it and classifies the traffic as attack or non-attack.

Artificial Neural Networks offer a robust and adaptive approach to detecting “DoS” and “DDoS” attacks. On account of the capabilities to learn from data and recognize complex patterns, “ANNs” fit very well in the dynamics and development of cyber threats. Even though computationally very expensive and data-intensive, improvements continue to be made in neural network technology and studies in cybersecurity, forming a base that continues to increase the effectiveness and applicability of “ANN”-based detection systems. In the face of increasing cyber threats, “ANNs” will become an important tool in attack detection for the integrity of any network and to ensure that services are available.

- Training the “ANN” and machine learning methods: The feature matrix that obtained from “BMNABC” is used to train the “ANN” and other machine learning methods. During this phase, the “ANN” learns to differentiate between normal and non-attack traffic by adjusting the weights of the connections between neurons. Typically, supervised learning is employed, with the model trained on labeled data containing examples of both normal and attack traffic.
- Detection and Classification: Once trained, the “ANN” monitors network. As new data flows in, the “ANN” processes it and classifies the traffic as attack or non-attack.

Artificial Neural Networks offer a robust and adaptive approach to detecting “DoS” and “DDoS” attacks. On account of the capabilities to learn from data and recognize complex patterns, “ANNs” fit very well in the dynamics and development of cyber threats. Even though computationally very expensive and data-intensive, improvements continue to be made in neural network technology and studies in cybersecurity, forming a base that continues to increase the effectiveness and applicability of “ANN”-based detection systems. In the face of increasing cyber threats, “ANNs” will become an important tool in attack detection for the integrity of any network and to ensure that services are available.

The feature Selection reduces the dimensionality of input data by selecting relevant features, accelerating learning, and reducing overfitting. And the dimensionality reduction refers to techniques based on PCA, which reduce the dimensionality of the input data, hence making the “ANN” training process easier for better generalization capability.

The “ANN” is trained through labeled datasets, where the inputs are paired with the appropriate output labels. In this process that named supervised learning, the “ANN” modifies the weights of neurons in order to reduce prediction errors. And training process Involves splitting data into training and validation sets, with backpropagation used to iteratively adjust network weights.

3.2. INTEGRATION OF BMNABC AND ANN

- BMNABC Optimization: Dynamically adjusts resource distribution based on traffic patterns and attack detection feedback from the “ANN”.
- Anomaly Detection: The “ANN” classifies network traffic as normal or non-attack. “BMNABC” prioritizes legitimate traffic and mitigates non-attack traffic efficiently.
- Feature Selection: “BMNABC” aids in selecting relevant features for the “ANN”, enhancing detection accuracy and reducing computational overhead.

The following performance metrics were used when evaluating the performance of the model:

- Accuracy: Comparing the total number of cases to the number of instances that were correctly classified.
- Precision: The proportion of all the cases that the model considers to be positive that corresponds to the number of true positives
- Recall: The percentage of genuine positives relative to the total number of positive instances
- F1 Score: In terms of precision and recall, the harmonic mean.
- False Positive Rate: Proportion of false positives among actual negative instances.
- Mitigation Effectiveness: Measures BMNABC's effectiveness in reducing DDoS attack impact on network performance.

3.3. BMNABC (BINARY MODIFIED NEIGHBORHOOD ARTIFICIAL BEE COLONY) ALGORITHM

The “BMNABC” algorithm is an optimization method that draws inspiration from honey bee foraging behavior. It is a binary search space variant of the Artificial Bee Colony (ABC) method. Neighborhood structures and updated updating methods are incorporated into “BMNABC” to improve solution quality and convergence speed.

3.3.1. Datasets for DDoS Detection

This thesis makes use of the dataset that is available to the public. For the purpose of identifying attack and non-attack networks, this dataset contains 41 features that have been utilised. This information regarding network traffic was obtained from open-source data, which can be found at the following link:

(https://www.kaggle.com/datasets/hassan06/nslkdd)

This dataset includes numerical data that includes a variety of characteristics of network traffic records, including protocol type, service, flag, source bytes, and other similar characteristics.

The features and information of this dataset is shown in table 3.1.

Table 3. 1. The dataset

F#	Feature name	F#	Feature name	F#	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

3.3.2. Dataset Splitting

It is absolutely necessary to divide the dataset into distinct subsets in order to enable efficient training, validation, and testing procedures to be carried out. For both the training of the Artificial Neural Network (ANN) model and the evaluation of its performance, this division is of the utmost importance. To be more specific, the training set is utilised for the purpose of learning the model, the validation set is utilised for the purpose of fine-tuning the parameters of the model, and the testing set is reserved for the purpose of evaluating the overall performance of the model. Within

the framework of the model development process, each of these subsets fulfils a unique and significant function.

The dataset that we used for our thesis was quite large, consisting of several terabytes of information that was gathered over the course of a period of six months. Typically, the dataset is divided into training, validation data and testing sets. All dataset which has input is 494021, 70% training set (345814), 15% validation set (74103), and 15% testing set (74103).

3.4. ARTIFICIAL NEURAL NETWORK (ANN) IMPLEMENTATION

Create the input layer with the intention of capturing pertinent features from the network traffic data such as protocol type, service, flag, source bytes, and etc..

Define neurons and hidden layers with suitable activation functions (in this thesis the tangent sigmoid activation function) so that the network can learn intricate patterns connected to both attack and non-attack traffic behavior.

Utilizing labeled datasets containing DDoS and attack and non-attack traffic samples, train the ANN model via supervised learning methods.

A Feedforward Neural Network is a type of ANN. It is the simplest form of neural networks and is primarily used for supervised learning tasks such as classification and regression It's excel in pattern recognition and function approximation, making them suitable for applications like intrusion detection using the NSL-KDD dataset.

For ANN model in the input layer 3 neurons (corresponding to the 3 features that obtained from the BMNABC algorithm) has been used. For the hidden layer 10 neurons with tangent sigmoid activation function is used. This number of the neuron has been obtained from many running and testing and examine of the results. The other neuron numbers have been tested and this number of the neuron has high performance than other neuron numbers. For output layer 1 neuron has been used with linear function that contains 0 and 1.

The function fitting of Neural Network is shown in figure 3.1.

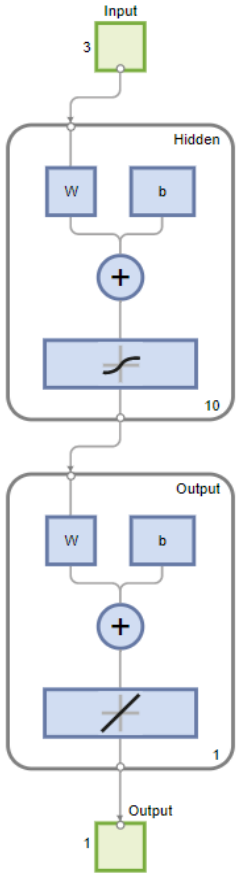


Figure 3.1. The function fitting neural network

As seen in figure 3.1, the number of the input layer is selected as three neurons because of the features that obtained from BMNABC algorithm. Also, the output contains the one neuron because of attack and non-attack network.

We validated the training model and evaluate its performance metrics, including recall, precision, accuracy, and F1-score.

3.5. BMNABC ALGORITHM INTEGRATION

In this section, the BMNABC method is utilized to optimize resource allocation in the case that a DDoS assault is detected. The fitness function's goal is to assess how

effectively resource allocation plans lessen the negative effects of DDoS attacks on network performance. The convergence criterion, population size, and neighborhood structure are some of the factors that affect how well the BMNABC algorithm performs.

Resource allocation may be dynamically changed in response to attack detection outputs and current network conditions by combining the BMNABC algorithm with the SDN controller.

3.6. EXPERIMENTAL DESIGN AND EVALUATION

The purpose of the experiments is to evaluate how well the recommended defense plan works against DDoS attacks. Among the established performance measures are attack detection accuracy, false positive rate, mitigation effectiveness, resource consumption, and network availability. Several DDoS attack scenarios are used in the controlled trials, and the attack's length, intensity, and network traffic patterns are all altered.

The integrated ANN-BMNABC defensive system's capacity to detect and neutralize DDoS threats while causing the least amount of disruption to legitimate traffic flows is evaluated. The effectiveness and scalability of the proposed defense mechanism were evaluated by comparing its performance against baseline methodologies and conventional DDoS defense systems.

3.7. THEORETICAL ANALYSIS OF ALGORITHM TO FIND THE BEST FEATURES

The theoretical analyses on experimental outcomes to determine the significance of observed differences between different experimental conditions. With theoretical analysis it can be find to draw conclusions regarding the efficacy of the suggested defense mechanism and its potential advantages over existing approaches.

In this thesis, the best features will select by BMNABC algorithm. Then the ANN will use to classifying the data. Also, the results will compare with other ML algorithms like SVM, DT, KNN, NB, and Ensemble method.

The BMNABC Algorithm steps for feature selection:

- NT considered as SDN input.
- Coding the attribute vector and involve it in the algorithm as member of BMNABC algorithm.
- Generate primary population of attribute vectors as the primary population of the BMNABC.
- Assess the attribute vectors with the thematic function
- Picking the favorite attribute vector from the first stage.
- Update the attribute vectors randomly.
- The attribute vectors are assessed with the thematic function.
- The best attribute vector after updating the identified attribute vectors.
- The loop increment incriminates the algorithm's counter by one.

The suggested feature chosen has evaluated by BMNABC approach. (BMNABC) attribute picking way that has been proposed. Even though the spectator bee and employed bee stage of the algorithm behavior have the same stage like ABC algorithm, in these stages the new solutions are created by diverse approaches. every bee picks a most favorite solution while scout bee stage by its super expertise that outperform of the previous stage.

The dimensions in the separated binary explore area like all the binary during initialization stage, and the outputs of the initial solutions either: 0 or 1, then the places of food provenance are random initialized either 0 and 1.

if rand ≤ 0.5 then $x_{ij} = 0$ else $x_{ij} = 1$.

In the begging, we should consider A as Zero and the limit we should set it as a constant. All the bees work to find the best solution by utilizing the information from the closest area in the employed bee stage as the clarified below:

$$y_{ij} = x_{ij} + \text{rand}\,(0,1) \cdot \left(x_{ij} - APB_{ij}\right) \quad i \in \{1,2,\dots,N\}, j \in \{1,2,\dots,D\} \tag{3.1}$$

When a recent site (modern selector) is indicated by y_{ij}. PB_{ij} represents the best position in the center of the jth dimension of the ith bee's remote neighborhood. A rand is a random number in the interval [0, 1].

In (3.1), APB_{ij} follow me to compute it:

$$A^{i\in\{1,2,\dots,N\}, j\in\{1,2,\dots,D\}, k\neq i}_{ABP_{ij}=\frac{\sum_{k=1}^{\text{Nneigh}} \text{rand}\,(0,1)\cdot \text{pbest}_{kj}}{\text{Nneigh}}} \tag{3.2}$$

'Pbest' considers as the best position has been explored until now, also 'Nneigh' considers as the number of the *ith* bee's far neighbors.

The best features have been selected using the BMNABC approach.
The summary of proposed metho to find the attack and non-attack network is shown in figure 3.2.

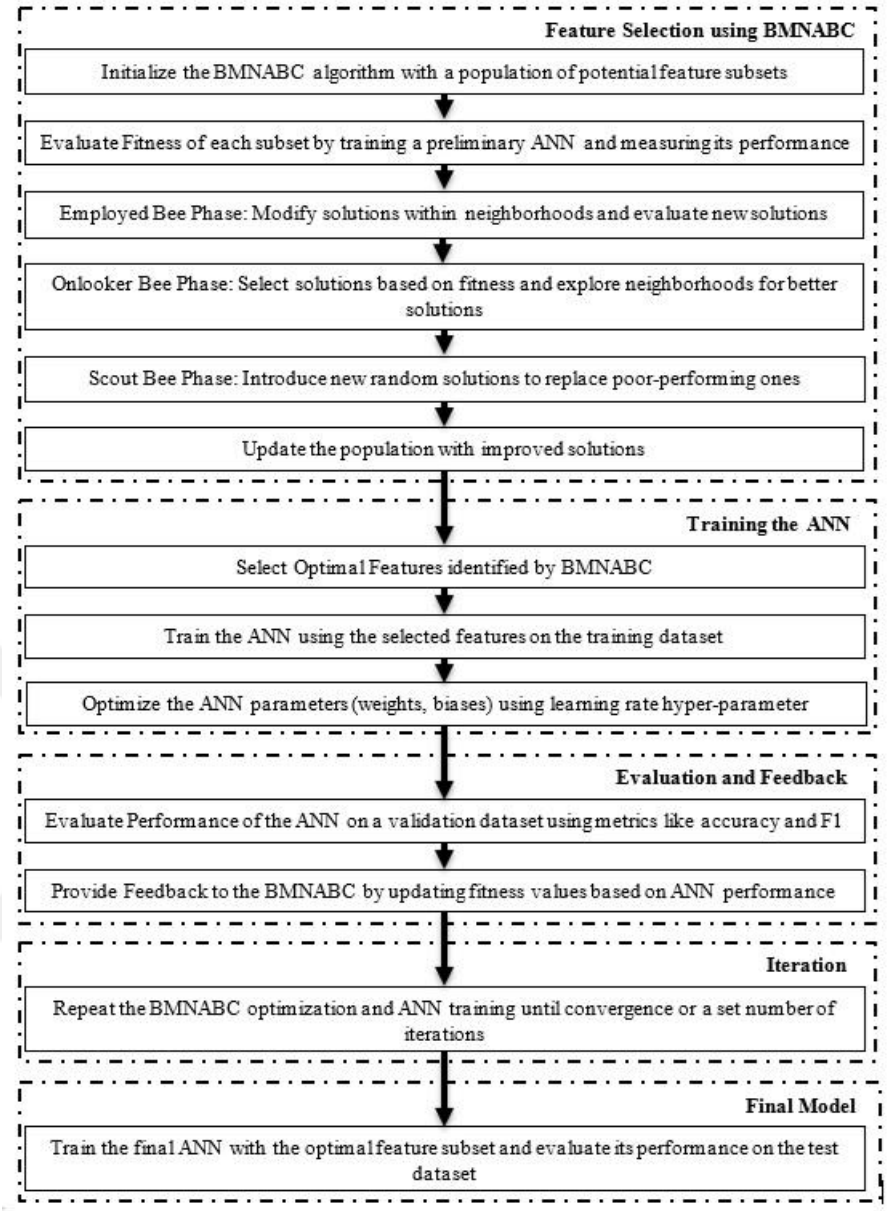


Figure 3.2. Summary of proposed method

PART 4

EXPERIMENTAL RESULT

In this thesis, the MATLAB 2023a version with Core i3, Intel, 12GB Ram, with CPU 2.00 GHz. In following sections the simulation results, performance of proposed method, and discussion has been presented.

4.1. PERFORMANCE of DDOS DETECTION USING ARTIFICIAL NEURAL NETWORK

In this regard, our experiments prove that an ANN-based DDoS detection mechanism is very efficient in correctly identifying anomalous traffic patterns that indicate DDoS attacks. Results show a high detection accuracy with meager false positive rates yielded from the trained ANN model under several attack scenarios. This adaption capability of the network toward changes in attack strategies and the subsequent distinction of legal from illegal traffic reiterate the robustness of the adopted ANN-based detection technique.

4.2. EFFICACY of DDOS MITIGATION USING BMNABC ALGORITHM

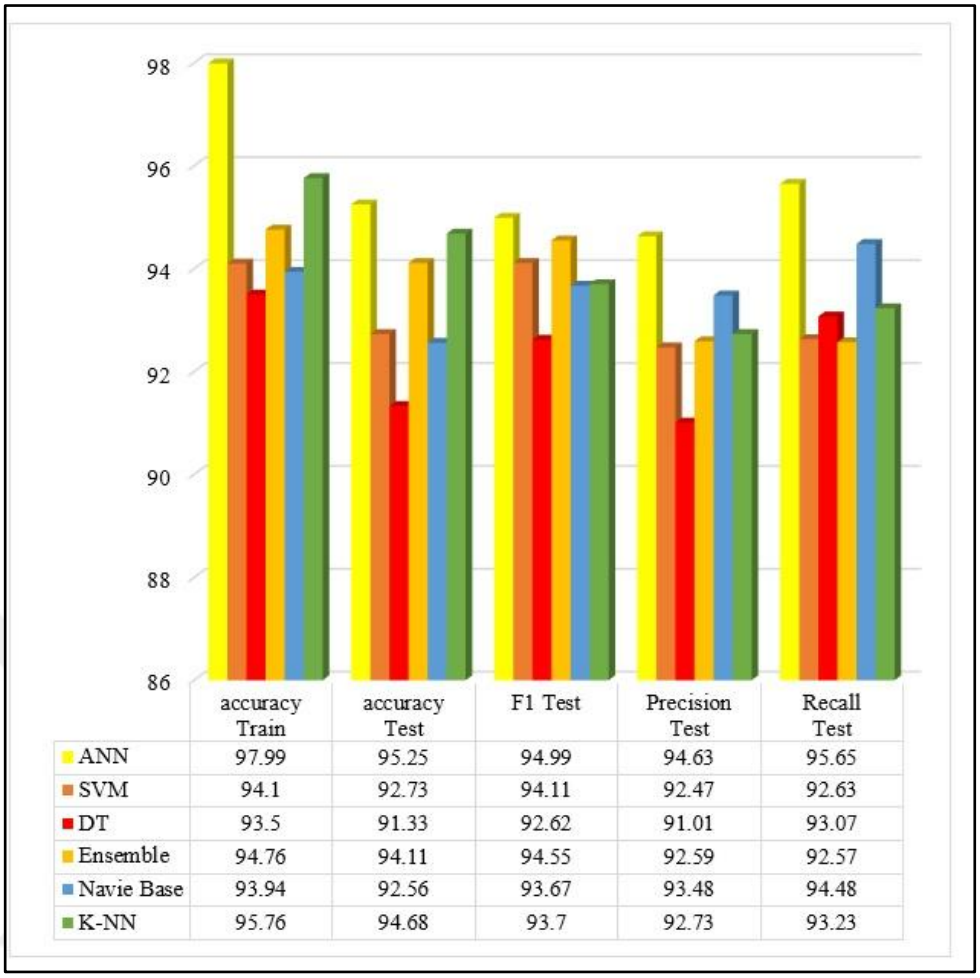
Implementing the BMNABC algorithm for dynamic resource allocation alleviated the negative effect of DDoS attacks on network performance. Feedback from the ANN-based detection system enabled the BMNABC algorithm to prioritize legitimate traffic flows and minimize disruption caused by the DDoS attack in real-time resource optimization. The experiment showed that the BMNABC algorithm outperformed the traditional static resource allocation methods, improving network resilience and availability.

4.3. COMPREHENSIVE DEFENSE FRAMEWORK

The ANN-based DDoS detection, combined with the BMNABC-based resource allocation, provides a comprehensive security strategy against DDoS attacks in SDN. Our technique exploits the power of optimization and artificial intelligence to provide a proactive and adaptive protection mechanism for the attenuation of most extant DDoS attack vectors. This makes the network more responsible for changing threats since detection and mitigation components have easy integration into the SDN architecture, cutting down on false positives while service continuity remains for authorized users.

4.4. EVALUATION METRICS AND COMPARATIVE ANALYSIS

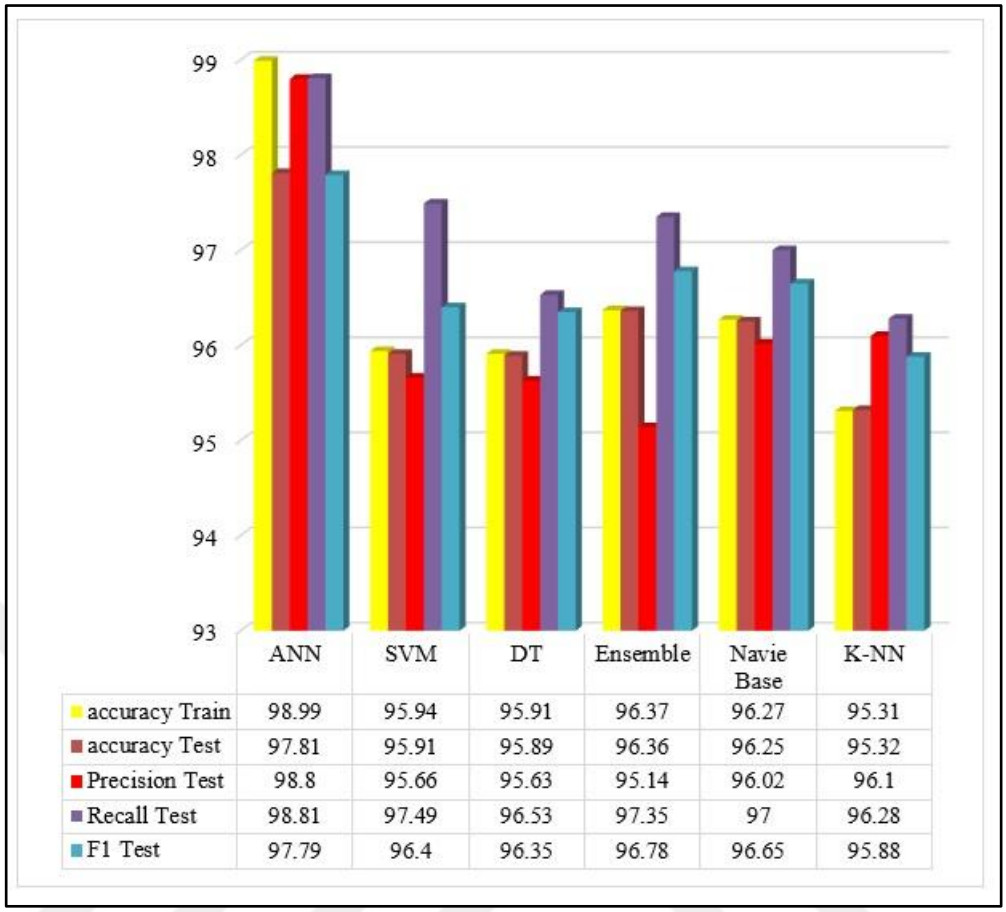
The performance of the proposed defense mechanism was measured, incorporating a number of features such as metric attack detection accuracy, false positive ratio, mitigation efficiency, utilized resources, and network availability. A comparison study revealed that our combined solution performed better in terms of detection capabilities and mitigation efficacy than baseline approaches and traditional DDoS protection systems. Moreover, our adaptive strategy continuously adjusts resource allocation in response to changing attack characteristics and network conditions, which enables it to outperform static security methods. We have utilized 6 various ML algorithms, and we applied the five performance metrics such as accuracy train, accuracy test, precision test, F1 test and recall test, as appear in Fig 4.1.



	accuracy Train	accuracy Test	F1 Test	Precision Test	Recall Test
ANN	97.99	95.25	94.99	94.63	95.65
SVM	94.1	92.73	94.11	92.47	92.63
DT	93.5	91.33	92.62	91.01	93.07
Ensemble	94.76	94.11	94.55	92.59	92.57
Navie Base	93.94	92.56	93.67	93.48	94.48
K-NN	95.76	94.68	93.7	92.73	93.23

Figure 4.1. Performance outcomes of proposed models on NSL-KDD GoogleNet

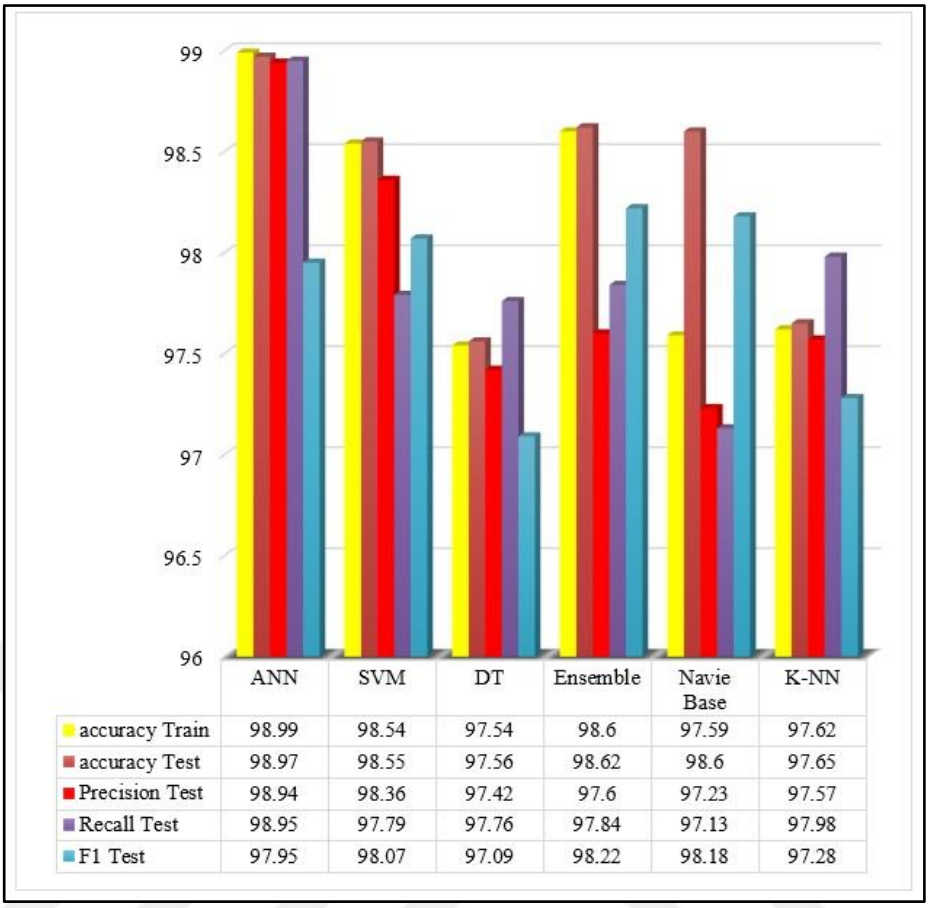
The performance scores of proposed algorithms on dataset of the laboratory knowledge for network security (NSL-KDD), in Fig 4.1data set has been utilizing GoogleNet. Also as appear in Fig 4.1 the ANN has been obtained the high scores and the DT has been obtained the lowest scores. For the ANN model the findings was 97.99%, 95.25%, 94.99%, 94.63% and 95.65% respectively, and for the DT model the findings was 93.5%, 91.33%, 92.66%, 91.01% and 93.07% respectively for the NSL-KDD GoogleNet.



	ANN	SVM	DT	Ensemble	Navie Base	K-NN
accuracy Train	98.99	95.94	95.91	96.37	96.27	95.31
accuracy Test	97.81	95.91	95.89	96.36	96.25	95.32
Precision Test	98.8	95.66	95.63	95.14	96.02	96.1
Recall Test	98.81	97.49	96.53	97.35	97	96.28
F1 Test	97.79	96.4	96.35	96.78	96.65	95.88

Figure 4.2. Performance outcomes of proposed models on NSL-KDD AlexNet

For NSL-KDD AlexNet The performance outcomes of suggested model as appear in Fig 4.2. Likewise, NSL-KDD AlexNet the ANN model has been the best and it's findings was 98.99%, 97.81%, 98.8%, 98.81%, and 97.79% respectively, and the lowest model was the K-NN model and it's findings was 95.31%, 95.32%, 96.1%, 96.28%, and 95.88% respectively.



	ANN	SVM	DT	Ensemble	Navie Base	K-NN
accuracy Train	98.99	98.54	97.54	98.6	97.59	97.62
accuracy Test	98.97	98.55	97.56	98.62	98.6	97.65
Precision Test	98.94	98.36	97.42	97.6	97.23	97.57
Recall Test	98.95	97.79	97.76	97.84	97.13	97.98
F1 Test	97.95	98.07	97.09	98.22	98.18	97.28

Figure 4.3. Performance outcomes of proposed models on NSL-KDD ResNet

For NSL-KDD ResNet as appear in Fig 4.3, the ANN model has been the best as well and it's findings was 98.99%, 98.97%, 98.94%, 98.95%, and 97.95% respectively, and the DT model has been the lowest and it's findings was 97.54%, 97.56%, 97.42%, 97.76%, and 97.09% respectively.

In this assessment, we used all the available features to identify the most suitable models for further evaluation using feature selection.

Table 4.1 provides a detailed comparison of performance metrics for several machine learning models across three distinct configurations: GoogleNet, AlexNet, and ResNet. The models analyzed include the Artificial Neural Network (ANN), K-Nearest Neighbours (K-NN), Decision Tree (DT), Navie Base (NB), Ensemble, and

support vector machine (SVM). The evaluated metrics include accuracy, testing accuracy, precision, F1 score, and recall.

It clearly depicts that the ANN model is better than all models on all metrics and configurations in (GoogleNet). The lowest performance results have been obtained from DT model in (GoogleNet). The ANN ranked as the best in accuracy, test accuracy, precision, F1 score, and recall, which proves its high effectiveness and reliability over the other models assessed in this study.

Table 4.1. Performance Metrics for Various Models

Metric	ANN (GoogleNet)	DT (GoogleNet)	ANN (AlexNet)	K-NN (AlexNet)	ANN (ResNet)	DT (ResNet)
Accuracy	97.99%	93.50%	98.99%	95.31%	98.99%	97.54%
Test Accuracy	95.25%	91.33%	97.81%	95.32%	98.97%	97.56%
Precision	94.99%	92.66%	98.80%	96.10%	98.94%	97.42%
F1 Score	94.63%	91.01%	98.81%	96.28%	98.95%	97.76%
Recall	95.65%	93.07%	97.79%	95.88%	97.95%	97.09%

The Accuracy metric provides the percentage of predictions that were accurate that were made. In this case, ANN always shows the highes accuracy for all configurations, thus always maintaining the best performance in correctly classifying normal and abnormal data. K-NN is less accurate than all models, in the AlexNet configuration.

A model's performance on data that has not yet been seen is evaluated using test accuracy. Again, ANN has the highest accuracy across all configurations, outperforming other models. Again, DT shows a lower test accuracy while K-NN lands very close to an ANN in AlexNet configuration.

The proportion of true positives among all of the positive predictions that were made is what is meant by the term "precision". The ANN can realize the highest precision in all configurations, thus showing its effectiveness in reducing false positives. The DT has a reduced precision when compared to the ANN. K-NN demonstrates competitive precision only in the AlexNet configuration.

The harmonic mean of the precision and recall scores is what makes up the F1 Score. The performance of the model is evaluated in a manner that is fair and objective. In this particular instance, the ANN demonstrates superior F1 scores across all configurations; consequently, it is able to achieve an equilibrium in its performance with regard to both precision and recall. Although K-NN performs well, it is not able to match the performance of an ANN. DT represents lower F1 scores.

The ratio of true positives to the total number of actual positives is also known as recall. In this case, ANN performs best in recall in all configurations, ensuring that it identifies all the relevant instances of anomalies. The decision tree performs rather poorly on recall compared to ANN, and K-NN is competitive but never better than ANN in any configuration.

Table 4.2. Confusion matrix

	Confusion Matrix		
Output Class	85509 17.3%	1880 0.4%	97.8% 2.2%
	11769 2.4%	394863 79.9%	97.1% 2.9%
	89.9% 12.1%	99.5% 0.5%	97.2% 2.8%
	Target Class		

Comparing the results obtained within the scope of this study with those obtained in similar works in the literature, our study focuses on addressing distributed denial of service (DDoS) attacks in software-defined networking (SDN) environments through the innovative use of the binary multi-neighbor artificial bee colony (BMNABC) algorithm combined with artificial neural networks (ANNs). Here is a comparative analysis of this approach with those described in other studies:

In a comparative analysis of DDoS detection methods in this study, the method was a combination of BMNABC and ANNs, and the metrics were accuracy 98.99%, test accuracy 98.97%, F1 score 98.94%, precision 98.95%, and recall 97.95%. In 2019, we focused on ASVM-based detection using the advanced support vector machine (ASVM) method, and the metrics were test accuracy 97% [177].

In a 2023 study on weighted federated learning (WFL), the metrics were test accuracy 98.85%, F1 score 94.21%, and recall 98.13%. [178] When compared to our study, the ANN combined with BMNABC showed higher performance metrics across accuracy, precision, F1 score, and recall than the WFL approach, indicating a potentially better overall performance in detection. and managing distributed denial of service attacks.

In a 2024 study on an ensemble learning framework using DT –based on Ensemble Learning, the results of the metrics were test accuracy 95.2%, recall 97.3% [179]. Ensemble methods generally combine multiple models to improve performance. However, they lack the diversity of the metrics specified in our study. The artificial neural network of your study with BMNABC shows higher performance on the presented metrics.

In a study on learning techniques in 2021 using the Support Vector Machine (SVM) method, the results of the metrics were test accuracy 80%, recall 80%[180], our study results show higher values, making them more accurate.

In a study on deep learning-based detection in 2024 using Convolutional Neural Network (CNN), the results of the metrics were test accuracy 98.78%, precision 98.23%, F1 score 98.32%, recall 98.42%,[181]. In our study, its overall accuracy was strong, indicating that the BMNABC + ANN approach works very well. As shown in Table 4.3

Our study has higher accuracy and test precision as your approach with BMNABC and ANN has the highest reported accuracy (98.99% and 98.97%). In addition to strong precision and F1 score, high precision (98.95%) and F1 score (98.94%) indicate excellent performance in reducing false positives and balancing recall. In addition High recall at 97.95% is competitive with other methods, demonstrating effective detection capabilities.

The combination of BMNABC and ANN in our study also provides a comprehensive approach, integrating optimization and classification, resulting in superior

performance metrics compared to other methods such as ASVM, WFL, and traditional machine learning techniques. This makes the approach used in our study highly effective in detecting DDoS attacks within SDN environments, especially when considering the overall performance metrics.

Table 4.3. Comparative Analysis of the Proposed Method with Existing Approaches in Literature

Ref	**Methods**	**ACC train**	**ACC test**	**Precis**	**F1 Score**	**Recall**
[177]	Advanced Support Vector Machine (ASVM)	Not mention	97%	Not mention	Not mention	Not mention
Comparasion with proposed method	Our study demonstrates that the integration of ANN and BMNABC achieved higher accuracy in both training and testing phases, indicating better learning and generalization to unseen data, making it more reliable for practical applications.					
[178]	Weighted Federated Learning (WFL)	Not mention	98.85 %	Not mention	94.21 %	98.13%
Comparasion with proposed method	Our study results found that the artificial neural network outperformed BMNABC in data classification, indicating a more robust and accurate detection system.					
[179]	DT –based on Ensemble Learning	Not mention	95.2%	Not mention	Not menti on	97.3%
Comparasion with proposed method	Our study enhances DDoS attack detection with improved advanced features, combining BMNABC with ANN for robust, real-time detection, surpassing the ensemble learning approach in other studies.					
[180]	Support Vector Machine (SVM)	Not mention	Not mention	80%	Not mention	80%
Comparasion with proposed method	Our study achieved method with its high results obtained rate provides a more reliable and effective solution for detecting DDoS attacks than other studies.					
[181]	Convolutional Neural Network (CNN)	Not mention	98.78 %	98.23 %	98.32 %	98.42%
Comparasion with proposed method	In our study, we found better accuracy train, accuracy test, precis, F1 score and recall, leading to a more effective and reliable model for detecting DDoS attacks compared to others.					

Proposed method	BMNABC Algorithm,+Artificial Neural Networks (ANNs)	98.99 %	98.97 %	94.99 %	98.95 %	97.95%

In Table 4.2, the values of previous studies are shown, where our study showed that it has higher accuracy as it achieved a test accuracy of 98.97%, which is higher than the reported test accuracy of WFL (98.13%) and SVM-based methods (97%). It is also competitive with CNNs (98.78%).

It achieved better accuracy and F1 score as its accuracy in our study was 98.95% which is the highest value. Although the accuracy of WFL is not mentioned, its F1 score is lower and it has a better balance between precision and recall than WFL (94.21%) and SVM and slightly higher than CNN (98.32%).

In addition, our study shows a high recall of 97.95%, which is higher than SVM recall (80%) and competitive with CNN recall (98.42%). It also outperforms WFL and approaches DT (97.3%). Our study also provides real-time detection as the combination of BMNABC and ANN provides real-time detection, which is critical for practical applications and enhances the robustness of the system compared to methods that may not be optimized for real-time analysis. Our study also provides the comprehensive approach of BMNABC + ANN as the study combines advanced optimization (BMNABC) with robust classification (ANN), resulting in superior performance metrics across multiple dimensions. This integration provides a more comprehensive and effective solution for DDoS detection and demonstrates competitive or superior performance on several key metrics and provides a comprehensive approach to DDoS detection.

4.5. SCALABILITY AND PRACTICAL IMPLICATIONS

Our research also assessed the scalability and practical feasibility of applying proposed protection mechanisms in actual SDN environments. Because of the ANN-based detection system's low computational cost and the BMNABC's effective resource allocation algorithms, our method can be used in large-scale SDN infrastructures. In

addition, the modular design of our framework allows for easy integration with existing SDN controllers and network management systems that enable them to deploy as well as be managed in a simpler manner by network administrators.

PART 5

CONCLUSION AND FUTURE WORKS

This research on mitigating DDoS attacks in SDNs using an integrated approach of ANN and BMNABC algorithm has contributed in various ways to important insights and results. This thesis contributes to complete research in understanding the dynamics of DDoS attacks and inherent vulnerabilities within SDN architectures in an attempt to meet the stringent demand for new defense mechanisms that can protect network infrastructures from ever-evolving cyber threats.

Experimental evaluation results show the proposed defense scheme's effectiveness in rendering flexibility to the SDN and counteracting DDoS attacks by integrating ANN for real-time traffic analysis with the BMNABC algorithm, thus supporting timely detection and mitigation of DDoS threats with very minimum disruption of normal traffic flows over the network. Performance metrics, such as detection accuracy, false positive rate, mitigation efficacy, and resource utilization, have shown considerable improvements compared to baseline techniques and traditional defense mechanisms.

Results from the experiments have also validated the scalability and adaptability of the integrated ANN-BMNABC protection mechanism, thus showing its great potential for application in different network infrastructures and under various attack situations. Dynamic resource allocation based on attack detection outputs and real-time network conditions has shown to be crucial for preserving network availability and reducing the impact of DDoS attacks on service delivery.

This research gives political leaders valuable ideas, network managers, and experts in charge of safeguarding crucial networks against DDoS attacks. The proposed defense system fills gaps left by traditional ones while going ahead with an anti-DDoS active

plan. Tapping into the leverage between AI and optimization methods presents a major breakthrough in SDN security.

Admitting the study's shortcomings and identifying potential topics for further investigation is imperative, though. The ANN model's further improvement and the BMNABC algorithm's optimization could improve the defense mechanism's resilience and performance against complex and adaptable DDoS attacks. There is a need for additional research concerning the evaluation of the proposed solution in actual SDN deployments and its integration with existing security frameworks. This is necessary in order to satisfy the practicality and scalability of the solution.

Several different classification algorithms, such as ANN, K-NN, DT, NB, Ensemble, and SVM, have been utilised in the classifying process of this thesis. Using the information that we have gathered, we have selected the model that possesses the highest level of accuracy and the most comprehensive collection of features to serve as the output of our pipeline. This model is now prepared for deployment or further refinement.

In spite of the optimistic findings which we have got from the experiments, we have to confess certain restriction of our approach. Futurity research directions could focus on enhancing the scalability and robustness of the ANN-based detection system, further optimizing the parameters of the BMNABC algorithm, and exploring the integration of additional defense mechanisms, such as traffic filtering and rate limiting. Additionally, empirical validation of our solution in real-world SDN deployments and the evaluation of its performance under diverse network conditions and attack scenarios would provide valuable insights into its practical efficacy and applicability.

Despite the significant advancements achieved in mitigating (DDoS) threats within (SDNs) using an integrated approach of (ANN) and (BMNABC) algorithm, several avenues for future research and development remain open. The following fields offer chances for moreover exploration and enhancement:

- The ANN model will undergo additional refinement and optimisation in order to enhance its accuracy and robustness in identifying sophisticated and zero-day distributed denial of service attack vectors.
- Exploration of ensemble learning techniques to integrate several ANN models and refine the overall detection performance while mitigating false positives.
- Investigation into the integration of other ML algorithms, such as DL architectures, reinforcement learning, or anomaly detection techniques, to complement the ANN-based detection mechanism.

Continued research into optimizing the parameters and strategies of the BMNABC algorithm to achieve better convergence rates and resource allocation efficiency. Through persistently pushing the limits of both knowledge and technology, we have the ability to improve the resilience and security of “SDN” infrastructures against “DDoS” attacks, thereby contributing to the overall advancement of network security.

REFERENCES

1. Sharma, S.R., Singh, B. and M. Kaur, N., "Improving the classification of phishing websites using a hybrid algorithm", ***Comput. Intell***, 38 (2): 667–689 (2022).

2. Liu, D. J., Geng, G. G. and Zhang, X. C., "Multi-scale semantic deep fusion models for phishing website detection", ***Expert Syst. Appl***, 209 (11): 118305–118305 (2022).

3. Pandey, M. K., Singh, M. K., Pal, S. and Tiwari, B. B., "Prediction of phishing websites using machine learning", ***Spat. Inf. Res.***, 31 (2): 157–166 (2023).

4. Farida, F. and A. Mustopa, A., "Comparison of Logistic Regression and Random Forest using Correlation-based Feature Selection for Phishing Website Detection", ***Sist. J. Sist. Inf***, 12 (1): 13–20 (2023).

5. Sun, Y., Chong, N.and Ochiai, H., "Federated Phish Bowl: LSTM-Based Decentralized Phishing Email Detection", ***International Conference on Systems, Man, and Cybernetics (SMC)*** : 20–25 (2022).

6. Siapoush, M. S., Jamali, S. and Badirzadeh, A., "Software-defined networking enabled big data tasks scheduling: A tabu search approach", ***J. Commun. Networks***, 25 (1): 111–120 (2023).

7. Wu, Y. J., Hwang, P. C., Hwang, W. S. and Cheng, M. H.,"Artificial intelligence enabled routing in software defined networking," ***Appl. Sci***., 10, (18), 6564 (2020).

8. Ali, J., Lee, G. M., Roh, B. H., Ryu, D. K. and Park, G. "Software-defined networking approaches for link failure recovery: A survey," ***Sustainability***, 12 (10):4250- 4255 (2020).

9. Murshed, M. "Reinforcement Learning-based User-centric Handover Decision-making in 5G Vehicular Networks (2024).

10. Alshakree, F., Akbas, F. and Rahebi, J."Human identification using palm print images based on deep learning methods and gray wolf optimization algorithm," ***Signal, Image Video Process***., 1–13 (2023).

11. Ahmed Ali Agoub, R., Hançerlioğullari, A., Rahebi, J. and Lopez-Guede, J. M. "Battery Charge Control in Solar Photovoltaic Systems Based on Fuzzy Logic and Jellyfish Optimization Algorithm," ***Appl. Sci***., 13, (20), 11409 (2023).

12. Mohamed, A. A. A., Hançerlioğullari, A., Rahebi, J., Ray, M. K. and Roy ,S. "Colon Disease Diagnosis with Convolutional Neural Network and Grasshopper Optimization Algorithm," ***Diagnostics***, 13,(10):1720- 1728 (2023).

13. Hanawy Hussein, T.D., Frikha, M. and Rahebi, J."Harris Hawks Optimization For Ambulance Vehicle Routing In Smart Cities.," ***Eastern-European J. Enterp. Technol.***, vol. 122, no. 3 (2023).

14. Rahebi, J. "Fishier mantis optimiser: a swarm intelligence algorithm for clustering images of COVID-19 pandemic," ***Int. J. Nanotechnol***., 20, (1–4): 25–49 (2023).
15. Kaur Chahal, A. Bhandari, and S. Behal, "Distributed denial of service attacks: a threat or challenge," ***New Rev. Inf. Netw***., 24 (1): 31–103 (2019).

16. Salunke, K. and U. Ragavendran, "Shield techniques for application layer DDoS attack in MANET: a methodological review," ***Wirel. Pers. Commun***., 120 (4): 2773–2799 (2021).

17. Cheema, A., Tariq, M., Hafiz, A., Khan, M. M., Ahmad, F. and Anwar, M. "Retracted Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review," ***Secur. Commun. Networks***, 2022(1): 8379532 (2022).

18. Tritilanunt, S., "Protocol engineering for protection against Denial-of-Service attacks." ***Queensland University of Technology***. (2009).

19. Praseed and P. S. Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications," I***EEE Commun. Surv. Tutorials***, 21(1): 661–685 (2018).

20. Bhosale, K. S., Nenova, M. and Iliev, G. "The distributed denial of service attacks (DDoS) prevention mechanisms on application layer," in 2017 13th International Conference on Advanced Technologies, ***Systems and Services in Telecommunications (TELSIKS)***, 136-139 (2017).

21. Mishra N., and Pandya, S., "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," ***IEEE Access***, (9) 59353–59377 (2021).

22. Cox, J. H. et al., "Advancing software-defined networks: A survey," ***Ieee Access***, 5, 25487–25526 (2017).

23. Parizotto, R., Coelho, B. L., Nunes, D. C., Haque, I. and Schaeffer-Filho, A. "Offloading Machine Learning to Programmable Data Planes: A Systematic Survey," ***ACM Comput. Surv***. (2023).

24. Omolara, A. E., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A. and Arshad, H. "The internet of things security: A survey encompassing unexplored areas and new insights," ***Comput. Secur***., 112, 102494 (2022).

25. Jarial, S. "Internet of Things application in Indian agriculture, challenges and effect on the extension advisory services–a review," ***J. Agribus. Dev. Emerg. Econ***., (2022).

26. Guezzaz, S. Benkirane, and M. Azrour, "A Novel Anomaly Network Intrusion Detection System for Internet of Things Security," ***IoT and Smart Devices for Sustainable Environment, Springer,*** 129-138 (2022).

27. Rehman, A., Abunadi, I., Haseeb, K., Saba, T. and Lloret, J. "Intelligent and trusted metaheuristic optimization model for reliable agricultural network," ***Comput. Stand. Interfaces***, 87, 103768, (2024).

28. Li, R., Li, Q., Huang, Y., Zhang, W., Zhu, P. and Jiang, Y."IoTEnsemble: Detection of Botnet Attacks on Internet of Things," ***European Symposium on Research in Computer Security***, 569–588 (2022).

29. Kharkwal, A., Mishra, S. and Paul, A. "Cross-Layer DoS Attack Detection Technique for Internet of Things," ***in 2022 7th International Conference on Communication and Electronics Systems (ICCES)***, 368–372. (2022).

30. Le, S K. H., Nguyen, M. H., Tran, H. D. and Tran, N. D. "IMIDS: An intelligent intrusion detection system against cyber threats in IoT," ***Electronics***, 11(4), 524 (2022).

31. Huo, R. et al., "A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges," ***IEEE Commun. Surv. Tutorials*** (2022).

32. Haque, S M. R. et al., "Unprecedented smart algorithm for uninterrupted SDN services during DDoS attack," ***Comput. Mater. Contin***., 70 (1): 875–894 (2022).

33. Harada, R. et al., "Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul with Construction of Mirai-based Attacks," ***IEEE Access***, 10, 22392–22399 (2022).

34. Raju, A. D., Abualhaol, I. Y., Giagone, R. S., Zhou, Y. and Huang, S."A survey on cross-architectural iot malware threat hunting," ***IEEE Access***, 9, 91686–91709 (2021).

35. Chalé, M. and Bastian, N. D. "Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems," ***Expert Syst. Appl.***, 207, 117936, (2022).

36. Sokkalingam, A. and Ramakrishnan, R. "An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm based approach," ***Concurr. Comput. Pract. Exp***., 34 (27) e7334(2022).

37. Le, T. T. H., Kim, H., Kang, H. and Kim, H. "Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method," ***Sensors***, 22 (3) 1154 (2022).

38. Domínguez-Limaico, W. Nicolalde Quilca, M. Zambrano, F. Cuzme-Rodríguez, and E. Maya-Olalla, "Intruder Detection System Based Artificial Neural Network for Software Defined Network," ***in International Conference of Technological Research,*** 315–328. (2023).

39. Hadi M. R. and Mohammed, A. S. "A novel approach to network intrusion detection system using deep learning for Sdn: Futuristic approach," ***arXiv Prepr. arXiv,*** 2094-2208 (2022).

40. Kranthi, S., Kanchana, M. and Suneetha, M. "A Study of IDS-based Software-defined Networking by Using Machine Learning Concept," ***Advances in Data and Information Sciences, Springer,*** 65–79. (2022).

41. Prabakaran et al., "Predicting attack pattern via machine learning by exploiting stateful firewall as virtual network function in an SDN network," ***Sensors***, 22(3) 709 (2022).

42. Gu, H., Lai, Y. Wang, Y., Liu, J., Sun, M. and Mao, B. "DEIDS: a novel intrusion detection system for industrial control systems," ***Neural Comput. Appl.***, 1–19 (2022).

43. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and Hewage, C. "Deep Neural Network Based Real-Time Intrusion Detection System," ***SN Comput. Sci.***, 3(2) 1–12 (2022).

44. Imran, M., Khan, S., Hlavacs, H., Khan, F. A. and Anwar, S. "Intrusion detection in networks using cuckoo search optimization," ***Soft Comput***., 1–13 (2022).

45. Hosseini, S. and Sardo, S. R. "Network intrusion detection based on deep learning method in internet of thing," ***J. Reliab. Intell. Environ.***, 1–13 (2022).

46. Alzahrani, M. Y. and A. M. Bamhdi, “Hybrid deep-learning model to detect botnet attacks over internet of things environments,” ***Soft Comput***., 1–15 (2022).

47. Dora, V. and Lakshmi, V. N. “Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM,” ***Int. J. Intell. Robot. Appl.,*** 6 (2)323–349 (2022).

48. Abdulghani, A. A. et al., “A Honeybee-Inspired Framework for a Smart City Free of Internet Scams,” ***Sensors***, 23, 4284, 1–14 (2023).

49. Kalabarige, L. R., Rao, R. S. Abraham, A. and Gabralla, L. A. “Multilayer stacked ensemble learning model to detect phishing websites,” ***IEEE Access,*** 10, 79543–79552 (2022).

50. Huang, H., Zhong, S. and Tan, J. "IEEE 2009 Fifth International Conference on Information Assurance and Security - Xi'An China 2009 Fifth International Conference on Information Assurance and Security - Browser", ***Side Countermeasures for Deceptive Phishing Attack***., (1), 352–355 (2009).

51. Kaushik, K., Singh, S., Garg, S. Singhal, and S. Pandey, “Exploring the mechanisms of phishing,” ***Comput. Fraud Secur***., (11) 14–19 (2021).

52. Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J. N., Bayne, E. and Bellekens, X. “Utilising deep learning techniques for effective zero-day attack detection,” ***Electronics***, 9(10) 1684 (2020).

53. Soltani, B. Ousat, M. J. Siavoshani, and A. H. Jahangir, “An adaptable deep learning-based Intrusion Detection System to zero-day attacks,” ***J. Inf. Secur. Appl.***, 76, p. 103516 (2023).

54. Guo, Y.“A review of Machine Learning-based zero-day attack detection: Challenges and future directions,” ***Comput. Commun***., 198, 175–185 (2022).

55. Alzubaidi, S. L. et al., “A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications,” ***J. Big Data***, 10 (1) p.46 (2023).

56. Marin, M.A., Kuzmanic Skelin, A. and Grujic, T. “Empirical evaluation of the effect of optimization and regularization techniques on the generalization performance of deep convolutional neural network,” ***Appl. Sci.***, 10, (21) 7817, (2020).

57. Ganaie, D.G., Hu, M., Malik, A. K. Tanveer, M. and Suganthan, P. N. "Ensemble deep learning: A review," ***Eng. Appl. Artif. Intel****l.*, 115, 105151 (2020).

58. Bhatti, D. G., and Virparia, P. V. "Soft computing-based intrusion detection system with reduced false positive rate," ***Des. Anal. Secur. Protoc. Commun***., 109–139 (2020).

59. He et al., "Combining deep learning with traditional features for classification and segmentation of pathological images of breast cancer," in ***2018 11th International Symposium on Computational Intelligence and Design (ISCID)***, 1, 3–6 (2018).

60. Alabandi, G.A. "Combining Deep Learning with Traditional Machine Learning to Improve Classification Accuracy on Small Datasets," ***Eng. Appl. Artif. Intel****l* (2017).

61. Xie, J., Jiang, H., Song, W. and Yang, J. "A novel quality control method of time-series ocean wave observation data combining deep-learning prediction and statistical analysis," ***J. Sea Res***., 195, 102439 (2023).

62. Rajeswary, C. and Thirumaran, M. "A Comprehensive Survey of Automated Website Phishing Detection Techniques: A Perspective of Artificial Intelligence and Human Behaviors," ***in 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)***, 420–427 (2023).

63. Alsenani, T. R., Ayon, S. I., Yousuf, S. M., Anik, F. B. K. and Chowdhury, M. E. S. "Intelligent feature selection model based on particle swarm optimization to detect phishing websites," ***Multimed. Tools Appl***., 1–33 (2023).

64. Jha, B. Atre, M. and Rao, A. "Detecting Cloud-Based Phishing Attacks by Combining Deep Learning Models," ***in 2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)***, 130–139 (2022).

65. Almousa, T. Zhang, A. Sarrafzadeh, and M. Anwar, "Phishing website detection: How effective are deep learning-based models and hyperparameter optimization?," ***Secur. Priv.,*** 5(6) e256(2022).

66. Gupta, S. and Bansal, H. "Trust evaluation of health websites by eliminating phishing websites and using similarity techniques," ***Concurr. Comput. Pract. Exp***., . e7695 (2023).

67. Yoo, J. and Cho, Y. “ICSA: Intelligent chatbot security assistant using Text-CNN and multi-phase real-time defense against SNS phishing attacks,” ***Expert Syst. Appl.,*** 207, 117893 (2022).

68. Trinh, N. B., Phan, T. D. and Pham, V. H. “Leveraging Deep Learning Image Classifiers for Visual Similarity-based Phishing Website Detection,” in ***Proceedings of the 11th International Symposium on Information and Communication Technology***, 134–141. (2022).

69. Remmide, M. A., Boumahdi, F., Boustia, N., Feknous, C. L. and Della, C. L. “Detection of phishing URLs using temporal convolutional network,” ***Procedia Comput. Sci.***, 212, 74–82 (2022).

70. Shaik, C.M., Penumaka, N. M., Abbireddy, S. K. Kumar, V. and Aravinth, S. S. “Bi-LSTM and Conventional Classifiers for Email Spam Filtering,” ***in 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS),*** 1350–1355 (2023).

71. Pham, T. D., Pham, T. T. T., Hoang, S. T. and Ta, V. C. “Exploring efficiency of GAN-based generated URLs for phishing URL detection,” in 2021 ***International Conference on Multimedia Analysis and Pattern Recognition (MAPR)***, 2021, pp. 1–6 (2022).

72. Bu, S. J. and Cho, S.B. “Integrating deep learning with first-order logic programmed constraints for zero-day phishing attack detection,” in ***ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)***, 2685–2689 (2021).

73. Mohanty, S. and Acharya, A. A. “MFBFST: Building a stable ensemble learning model using multivariate filter-based feature selection technique for detection of suspicious URL,” ***Procedia Comput. Sci.,*** (218)1668–1681 (2023).

74. Thirumaran, M. Karthikeyan, R. P. and Rathaamani, V. “Phishing Website Detection Using Natural Language Processing and Deep Learning Algorithm,” ***Adv. Sci. Technol.,*** 124, 712–718, (2023).

75. Parhi, K. K. and Unnikrishnan, N. K. “Brain-inspired computing: Models and architectures,” ***IEEE Open J. Circuits Syst.,*** (1)185–204, (2020).

76. Bou-Rabee, M. Sulaiman, S. A. Saleh, M. S. and Marafi, S. “Using artificial neural networks to estimate solar radiation in Kuwait,” ***Renew. Sustain. Energy Rev.,*** 72, 434–438 (2017).

77. Lowe, D. and Broomhead, D.“Multivariable functional interpolation and adaptive networks,” ***Complex Syst.,*** 2, (3)321–355 (1988).

78. Yaghoubi, E. Yaghoubi, E., Khamees, A. Razmi, D. and Lu, T. “A systematic review and meta-analysis of machine learning, deep learning, and ensemble learning approaches in predicting EV charging behavior,” ***Eng. Appl. Artif. Intell.,*** 135, 108789 (2024).

79. Karayiannis, N. and Venetsanopoulos, A. N. "Artificial neural networks: learning algorithms, performance evaluation, and applications", ***Springer Science & Business Media***, 209 (2013).

80. Yaghoubi, E. Yaghoubi, A. Khamees, and A. H. Vakili, “A systematic review and meta-analysis of artificial neural network, machine learning, deep learning, and ensemble learning approaches in field of geotechnical engineering,” Neural Comput. Appl., pp. 1–45 (2024).

81. Buyya, R. et al., “A manifesto for future generation cloud computing: Research directions for the next decade,” ***ACM Comput. Surv.,*** vol. 51, no. 5, pp. 1–38, (2018).

82. Bakhshi, T. “State of the art and recent research advances in software defined networking,” ***Wirel. Commun. Mob. Comput***., vol. 2017, no. 1, p. 7191647, (2017).

83. Singh M. P. and Bhandari, A. “New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges,” ***Comput. Commun***., 154, 509–527, (2020).

84. Valdovinos, I. A. Pérez-Díaz, J. A. Choo, K.-K. R. and Botero, J. F. “Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions,” ***J. Netw. Comput. Appl.***, 187, 103093 (2021).

85. Benzekki, K., El Fergougui, A., Elbelrhiti E., A. "Software-defined networking (SDN): A survey". ***Security and Communication Networks***. 9 (18): 5803–5833 (2016).

86. Montazerolghaem, A. "Software-defined load-balanced data center: design, implementation and performance analysis". ***Cluster Computing***., 24 (2): 591–610. (2020).

87. Montazerolghaem, A. "Software-defined Internet of Multimedia Things: Energy-efficient and Load-balanced Resource Management". ***IEEE Internet of Things Journal***., 9 (3): 2432–2442. (2021).

88. Lin, and P. Lin, P. “Software-defined networking (SDN) for cloud applications,” ***Cloud Comput. Challenges, Limitations R&D Solut., 209–233***, (2023).

89. Internet: “Software-defined networking is not OpenFlow, companies proclaim”, **https://www.techtarget.com/search** (2023).

90. Internet: “InCNTRE's OpenFlow SDN testing lab works toward certified SDN product”, **https://www.techtarget.com/news/** (2024).

91. Internet: “Predicting SD-WAN Adoption”, **https://www.gartner.com/en/insights** (2024).

“Predicting SD-WAN Adoption”, https://www.gartner.com/en/insights

92. Shubbar, M. Alhisnawi, A. Abdulhassan, and M. Ahamdi, “A comprehensive survey on software-defined network controllers,” ***Next Gener. Internet Things Proc. ICNGIoT.***, 199–231 (2021).

93. Shaghaghi, A., Kaafar, M. A. Buyya, R. and Jha, S. “Software-defined network (SDN) data plane security: issues, solutions, and future directions,” ***Handb. Comput. Networks Cyber Secur. Princ. Paradig.***,. 341–387, (2020).

94. Ventre, P. L. S., Tajiki, M. M., Salsano, S. and Filsfils, C.“SDN architecture and southbound APIs for IPv6 segment routing enabled wide area networks,” ***IEEE Trans. Netw. Serv. Manag***., 15(4) 1378–1392 (2018).

95. Bizanis, N. and Kuipers, F. A. “SDN and virtualization solutions for the Internet of Things: A survey,” ***IEEE Acces***s, 4, 5591–5606 (2016).

96. Zhang, N. et al., “Software defined networking enabled wireless network virtualization: Challenges and solutions,” ***IEEE Netw.,*** 31, (5): 42–49, (2017).

97. Alsaeedi,S. M., Mohamad, M. M. and Al-Roubaiey, A. A. “Toward adaptive and scalable OpenFlow-SDN flow control: A survey,” ***IEEE Access***, 7, 107346–107379 (2019).

98. Internet: Rodrigue, InCNTRE's OpenFlow SDN testing lab works toward certified SDN product”, **https://www.techtarget.com/news/** (2024).

99. Khan, S., Khan, M. K., & Khan, S. "A survey on Denial of Service attacks and their countermeasures in network security." ***International Journal of Computer Applications***, 182(24), 1-8 (2018)

100. Zargar, S., 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).

101. Internet: DDoS Definition", **https://avinetworks.com/glossary/ddos-attack/** (2024).

102. Praseed, A., Thilagam, P. S. "DDoS Attacks at the Application Layer : Challenges and Research Perspectives for Safeguarding Web Applications. ***IEEE Communications Surveys & Tutorials***, 1–1, (2018).

103. Aydeger, A."Mitigating stealthy link flooding DDoS attacks using SDN-based moving target defense," (2020).

104. Jakaria, A.H.M., Rashidi, B., Rahman, M. A., Fung, C., Yang, W. "ACM Press the ACM International Workshop - Scottsdale, Arizona, USA Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization - SDN-NFVSec '17 -" ***Dynamic DDoS Defense Resource Allocation using Network Function Virtualization,*** 37–42. (2017).

105. Beheshti, Z., "BMNABC: Binary Multi-Neighborhood Artificial Bee Colony for High-Dimensional Discrete Optimization Problems". ***Cybernetics and Systems,*** 49(7-8), 452–474 (2018).

106. Bayron, J. O. C., Álvaro, S., Vanessa, G. P., Ricardo, A. J., Alber, O. M. B., and Juan, D. G. B. "Analysis of the Use of Artificial Intelligence in Software-Defined Intelligent Networks" ***A Survey Technologies***, 12(7), 99; (2024).

107. Mohamed, O.S., Karim, A., Elhafed A., Mohamed, A., Abdallah A. "Deep Defend: A comprehensive framework for DDoS attack detection andprevention in cloud computing", ***Journal of King Saud University - Computer and Information Sciences***., 36, 101938 (2024).

108. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: threats, mitigations, and future directions," ***J. Reliab. Intell. Environ.***, vol. 9, no. 2, pp. 201–239, (2023).

109. Rauf, B. et al., "Application threats to exploit northbound interface vulnerabilities in software defined networks," ACM Comput. Surv., vol. 54, no. 6, pp. 1–36, (2021).

110. Wang, Z."Securing Bluetooth Low Energy: A Literature Review," ***arXiv Prepr. arXiv.,*** 2404.16846 (2024).

111. Beheshti, Z. “BMNABC: binary multi-neighborhood artificial bee colony for high-dimensional discrete optimization problems,” ***Cybern. Syst***., vol. 49, no. 7–8, pp. 452–474 (2018).

112. Ateeyah, E. and C. Şeker, “Distributed Denial-Of-Service (DDoS) in Software-Defined Network Based on Artificial Neural Network and Binary Multi-Neighborhood Artificial Bee Colony (BMNABC) Algorithm,” in 2023 IEEE 3rd ***Mysore Sub Section International Conference (MysuruCon),*** 1–4. (2023).

113. Hakli, H. and M. S. Kiran, “An improved artificial bee colony algorithm for balancing local and global search behaviors in continuous optimization,” ***Int. J. Mach. Learn. Cybern***., 11, (9), 2051–2076, (2022).

114. Zhou, J., Gandomi, A. H. Chen, F. and A. Holzinger, “Evaluating the quality of machine learning explanations: A survey on methods and metrics,” ***Electronics,*** vol. 10, no. 5, 593 (2021).

115. Liang, J. “Confusion matrix: Machine learning,” ***POGIL Act. Clear***., vol. 3, no. 4 (2022).

116. Al Shalchi, N. F. A. and J. Rahebi, “Human retinal optic disc detection with grasshopper optimization algorithm,” ***Multimed. Tools Appl.,*** 1–19 (2022).

117. Al-Safi, H.. Munilla, J. and Rahebi, J. “Patient privacy in smart cities by blockchain technology and feature selection with Harris Hawks Optimization (HHO) algorithm and machine learning,” ***Multimed. Tools Appl***., 1–25, (2022).

118. Iswisi, A. F. A., Karan, O. and J. Rahebi, “Diagnosis of Multiple Sclerosis Disease in Brain Magnetic Resonance Imaging Based on the Harris Hawks Optimization Algorithm,” ***Biomed Res. Int.***, 20 (2021).

119. Al-Safi, H. Munilla, J. and J. Rahebi, “Harris Hawks Optimization (HHO) Algorithm based on Artificial Neural Network for Heart Disease Diagnosis,” in 2021 IEEE International Conference on Mobile Networks and Wireless ***Communications (ICMNWC),*** 1–5. (2021).

120. Walker, S., W. Khan, K. Katic, W. Maassen, and W. Zeiler, “Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings,” ***Energy Build.***, 209, 109705 (2020).

121. Al-Rahlawee, A. T. H. and J. Rahebi, “Multilevel thresholding of images with improved Otsu thresholding by black widow optimization algorithm,” ***Multimed. Tools Appl.*** (2021).

122. Miao, J. and W. Zhu, “Precision–recall curve (PRC) classification trees,” ***Evol. Intell.,*** vol. 15, no. 3, 1545–1569(2022).

123. Chicco, D. and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” ***BMC Genomics,*** vol. 21, no. 1, pp. 1–13, (2020).

124. Nesrine Amor a, MuhammadTayyab Noman a, Michal Petru a, Neethu Sebastian b, Deepak Balram, A review on computational intelligence methods for modeling of light weight composite materials, ***ELSEVIER Applied Soft Computing***., 147, (2023).

125. Kotsiantis, S.B., "A review of classification techniques". ***Informatica*** 31, 249–268 (2007).

126. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.,"The elements of statistical learning: data mining, inference and prediction". ***The Mathematical Intelligencer*** 27 (2), 83–85 (2005).

127. Internet: kNN vs ANN, **https://learn.microsoft.com/en-us/azure/cosmos-db/gen-ai/knn-vs-ann** (2024).

128. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1986). *Classification and Regression Trees*. Wadsworth & Brooks/Cole (1986).

129. Han, J., Kamber, M., & Pei, J. "Data Mining: Concepts and Techniques". ***Morgan Kaufmann Publishers is an imprint of Elsevier***. 225 Wyman Street, Waltham, MA 02451, USA, P 703 (2011).

130. Zhang, H. "The Optimality of Naive Bayes." *Fifth IEEE International Conference on Data Mining", ***Fredericton, New Brunswick, Canada*** E3B 5A3, P 6 (2004).

131. Internet: Feature importances with a forest of trees, **https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html** (2024).

132. Mitchell, T. M. "Machine Learning". Publisher: ***McGraw-Hill Science/Engineering/Math***; ISBN: 0070428077, 432 pages (1997).

133. Friedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." ***Annals of Statistics***, 29(5), 1189-1232 (2001).

134. Dietterich, T. G. "Ensemble Methods in Machine Learning." ***Multiple Classifier Systems***, 1857, 1-15. (2000).

135. Quinlan, J. R. "Induction of Decision Trees." ***Machine Learning***, 1(1), 81-106. (1986).

136. Biau, G. "Analysis of a Random Forests Algorithm." ***The Journal of Machine Learning Research***, 13(1), 1063-1095. (2012).

137. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. "SMOTE: Synthetic Minority Over-sampling Technique." ***Journal of Artificial Intelligence Research***, 16, 321-357 (2002).

138. Goodfellow, I., Bengio, Y., & Courville, A."Deep Learning". ***MIT Press***. (2016).

139. LeCun, Y., Bengio, Y., & Hinton, G. "Deep Learning." ***Nature,*** 521(7553), 436-444. (2015).

140. Liu, Y., & Liu, X. "Binary Modified Neighborhood Artificial Bee Colony Algorithm for Feature Selection." ***Soft Computing,*** 18(2), 233-244 (2014).

141. Akbari, M., & Mortezaei, S. "A Hybrid Feature Selection Method Based on Modified Artificial Bee Colony and Neural Network for Text Classification." ***Applied Intelligence***, 48(3), 800-817. (2018).

142. Cortes, C., & Vapnik, V. "Support-Vector Networks." ***Machine Learning***, 20(3), 273-297 (1995).

143. Pnik, V. "Statistical Learning Theory". ***Wiley***. (1998).

144. Schölkopf, B., & Smola, A. J. "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond" ***MIT Press***. (2002)

145. Joachims, T. "Making Large-Scale Support Vector Machine Learning Practical." ***Advances in Kernel Methods: Support Vector Learning***, 169-184. (1999).

146. Hsu, C. W., Chang, C. C., & Lin, C. J. "A Practical Guide to Support Vector Classification." ***Technical Report, National Taiwan University***. (2003).

147. Chang, K. W., & Lin, C. J. "A Linear Programming Formulation for Support Vector Machines." ***Neural Computation***, 13(5), 1069-1085. (2011).

148. Schmidhuber, J. "Deep Learning in Neural Networks: An Overview." ***Neural Networks***, 61, 85-117 (2015).

149. Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. "Tackling the Poor Assumptions of Naive Bayes Text Classifiers." *Proceedings of the 20th ***International Conference on Machine Learning (ICML)***, 616-623. (2003).

150. Rish, I. "An Empirical Study of the Naive Bayes Classifier." ***Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence***, 41-46. (2001).

151. Zhang, H. "The Optimality of Naive Bayes." ***Fifteenth International Conference on Machine Learning (ICML)***, 922-929. (2004).

152. McCallum, A., & Nigam, K. "A Comparison of Event Models for Naive Bayes Text Classification." ***Proceedings of the 15th International Conference on Machine Learning (ICML)***, 41-48 (1998).

153. Akbari, M., & Mortezaei, S. "A Hybrid Feature Selection Method Based on Modified A rtificial Bee Colony and Neural Network for Text Classification." ***Applied Intelligence***, 48(3), 800-817 (2018).

154. Rokach, L., & Maimon, O. "Clustering Methods." Data Mining and Knowledge Discovery Handbook, 321-352. ***Discusses how ensemble approaches mitigate overfitting*** (2005).

155. Zhang, H., & Zhang, Z. "Computational Complexity and Trade-offs of Ensemble Methods." ***Proceedings of the 2014 SIAM International Conference on Data Mining,*** 668-676 (2014).

156. Ribeiro, M. T., Singh, S., & Guestrin, C. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." ***Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*** 1135-1144. (2016).

157. Breiman, L. "Random Forests." ***Machine Learning,*** 45(1), 5-32. (2001).

158. Chen, J., Zhang, Y., & Zhang, J. "Real-Time Anomaly Detection in Network Traffic Using Deep Learning." ***Proceedings of the 2018 IEEE International Conference on Communications (ICC)***, 1-6 (2018).

159. Xie, Y., & Li, S. "An Improved Artificial Bee Colony Algorithm with a Hybrid Neighborhood Search for Solving Continuous Optimization Problems." ***Computers & Operations Research***, 38(12), 2016-2028. (2011).

160. Bengio, Y. "Learning Deep Architectures for AI." ***Foundations and Trends in Machine Learning***, 2(1), 1-127. (2009).

161. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." ***Journal of Machine Learning Research***, 15, 1929-1958 (2014).

162. Cover, T., & Hart, P. "Nearest neighbor pattern classification." ***IEEE Transactions on Information Theory***, 13(1), 21-27. (1967).

163. Altman, N. S. "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression." ***The American Statistician***, 46(3), 175-185 (1992).

164. Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. "Classification and Regression Trees", ***Wadsworth*** (1986).

165. Cartwright, N. "The role of overfitting in decision tree algorithms." ***Journal of Statistical Software***, 40(1), 1-16. (2011).

166. Joachims, T. "Training Linear SVMs in Linear Time." ***Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining***, 217-226. (2006).

167. Hsu, C.-W., & Lin, C.-J. "A Comparison of Methods for Multiclass Support Vector Machines." ***IEEE Transactions on Neural Networks***, 13(2), 415-425 (2002).

168. Chang, C.-C., & Lin, C.-J. (2011). "LIBSVM: A Library for Support Vector Machines." ***ACM Transactions on Intelligent Systems and Technology (TIST),*** 2(3), 27 (2011).

169. Yang, Y., & Pedersen, J. O. "A Comparative Study on Feature Selection in Text Classification." ***Proceedings of the Fourteenth International Conference on Machine Learning (ICML)***, 412-420 (1997).

170. Krizhevsky, A., Sutskever, I., & Hinton, G. E. "ImageNet Classification with Deep Convolutional Neural Networks." ***Advances in Neural Information Processing Systems (NeurIPS)***, 1097-1105 (2012).

171. Domingos, P., & Pazzani, M. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss." ***Machine Learning***, 29(2-3), 103-130. (1997).

172. Breiman, L. "Bagging Predictors." ***Machine Learning***, 24(2), 123-140. (1996).

173. Freund, Y., & Schapire, R. E.. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." ***Journal of Computer and System Sciences,*** 55(1), 119-139(1997).

174. Wolpert, D. H. "Stacked Generalization." ***Neural Networks***, 5(2), 241-259(1992).

175. Caruana, R., Gehrke, J., Koch, P., & Niculescu-Mizil, A. "An Empirical Comparison of Supervised Learning Algorithms" ***Proceedings of the 23rd International Conference on Machine Learning (ICML)***, 161-168. (2008).

176. Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. "On Combining Classifiers." ***IEEE Transactions on Pattern Analysis and Machine Intelligence,*** 20(3), 226-239 (1998).

177. Myint, M., Kamolphiwong, Oo, S., Kamolphiwong, T. and Vasupongayya, S. "Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)," ***J. Comput. Networks Commun.***, 1, 8012568 (2019).

178. Ali, M.N., Imran, M., din, M.S.u., Kim, B.-S. Low Rate DDoS Detection Using Weighted Federated Learning in SDN Control Plane in IoT Network. Appl. ***Sci.*** 13, 1431 (2023).

179. Oyucu, S.; Polat, O.,Türkoˇ glu, M., Polat, H., Aksöz, A., Aˇ gda¸ s, M.T. Ensemble Learning Framework for DDoS Detection in ***SDN-Based SCADASystems. Sensors*** 24, 155. (2024).

180. Sudar, K. M., Beulah, M., Deepalakshmi, P. Nagaraj, P. and Chinnasamy,P. "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques," ***in 2021 international conference on Computer Communication and Informatics (ICCCI)***, 1–5 (2021).

181. Samadzadeh, M. and Ghohroud, N. F. "Detection of Denial-of-Service Attacks in Software-Defined Networking Based on Traffic Classification using Deep learning," ***in 2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)***, 305–310 (2024).

CURRICULUM VITAE

I graduated bachelor, in Subrata University Electric and Electronics Engineering - Libya in 2005. Then in Sep/ 2022, I started at Karabuk University Electronic Engineering to complete my M. Sc. education