Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning

Thesis · May 2023

DOI: 10.13140/RG.2.2.33736.12803

Nisarg Mehta
Pandit Deendayal Petroleum University

Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning

A thesis submitted to

in partial fulfillment of the requirements for the degree of

Master of Technology

in

Cyber Security

by

Nisarg Mehta

21MCS003

Under the Guidance of
Prof. Samir Patel & Darshit Shah

Department of Computer Science & Engineering,

School of Technology, Pandit Deendayal Energy University,

Gandhinagar 382 426

May 2022

Certificate of Originality of Work

I hereby declare that the MTech. Project entitled “. Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning” submitted by me for the partial fulfillment of the degree of Master of Technology to the Dept. of Computer Science & Engineering at the School of technology, Pandit Deendayal Energy University, Gandhinagar, is the original record of the project work carried out by me under the supervision of Prof. Samir Patel & Darshit Shah

I also declare that this written submission adheres to university guidelines for its originality, and proper citations and references have been included wherever required.

I also declare that I have maintained high academic honesty and integrity and have not falsified any data in my submission.

I also understand that violation of any guidelines in this regard will attract disciplinary action by the institute.

Name of the Student:

Roll Number of the Student:

Signature of the Student:

Name of the Supervisor:

Designation of the Supervisor:

Signature of the Supervisor:

Place:

Date:

Certificate from the Project Supervisor/Head

This is to certify that the "Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI" submitted by Mr. Nisarg Mehta, Roll No. 21MCS003 towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Computer Science & Engineering from the School of Technology, Pandit Deendayal Energy University, Gandhinagar is the record of work carried out by him under our supervision and guidance. The work submitted by the student has in our opinion reached a level required for being accepted for examination. The results embodied in this Master Thesis work to the best of our knowledge have not been submitted to any other University or Institution for the award of any degree or diploma.

Name and Sign of the Supervisor 1

Name and Sign of the Supervisor 2

Name and Sign of the HoD

Name and Sign of the Director

Place

Date

Acknowledgment

In the very beginning, my thanks, and praises to God, the Almighty, for the choicest blessings showered on me to enable me to complete the research work successfully.

I am indebted to my supervisor Dr. Samir Patel, for invaluable guidance and input in the work. I am also thankful to my other supervisor Mr. Darshit Shah for kind advice and support. Their immense knowledge, patience and inspiration helped me to turn the disjointed ideas into one integrated and consistent research work. It was a great privilege to work and learn under their august guidance.

I am also grateful to Dr. Shakti Mishra, Head of the Department, Computer Science and Engineering, for the innumerable discussions that enriched the work.

Words are simply not enough to express enough gratitude to Dr. Madhu Shukla, and Dr. Pooja Shah, Mr. Nilesh Pandey, and Mr. Milind Shah for their constant motivation.

(Nisarg Mehta)

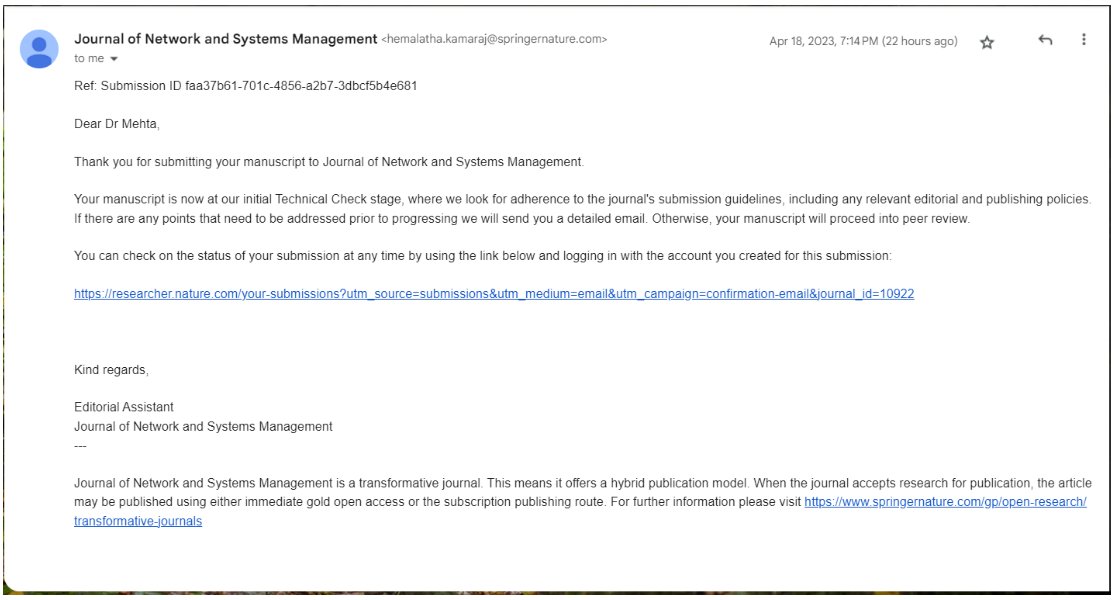
Abstract

Distributed Denial-of-Service (DDoS) infiltrations being an escalating menace to Software-Defined Networking (SDN) infrastructure. It is vital to have effective mechanisms in place to predict and counter these cyberattacks. In this thesis, we recommend an innovative methodology to anticipate DDoS incursions in SDN using Explainable AI (XAI) and Federated Learning (FL). Our novel framework employs XAI techniques to enable network administrators to decipher the results of the prediction with precision. FL is integrated to train machine learning models using data from diverse sources, while prioritizing data privacy to safeguard confidential network information. Additionally, we explore various XAI techniques and analyze their effectiveness in interpreting the results of our models. Our proposed framework was evaluated using real-world network traffic datasets, and its performance was compared with existing techniques. Results revealed that the amalgamation of XAI and FL techniques surpassed current methods in terms of predictability and interpretability. We demonstrated the effectiveness of different XAI techniques in producing explicable results that can assist network administrators in identifying the causes of an attack and formulating effective countermeasures. Overall, our innovative framework provides a promising solution for anticipating DDoS incursions in SDN infrastructure. It serves as a valuable tool for network administrators in detecting and neutralizing cyberattacks. The framework employs XAI and FL techniques to achieve precise, efficient, and explicable DDoS incursion anticipation in SDN. This thesis contributes to the field of SDN security by proposing a solution that can safeguard SDN networks from DDoS incursions.

Contents

LIST OF PUBLICATIONS

1. Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning by Nisarg Mehta, Dr.Madhu Shukla, Dr. Pooja Shah, Dr. Samir Patel Mr. Darshit Shah, Mr. Kishan Makdiya submitted to Journal of Network and Systems Management.



Journal of Network and Systems Management <hemalatha.kamaraj@springernature.com>

to me

Apr 18, 2023, 7:14 PM (22 hours ago)

Ref: Submission ID faa37b61-701c-4856-a2b7-3dbcf5b4e681

Dear Dr Mehta,

Thank you for submitting your manuscript to Journal of Network and Systems Management.

Your manuscript is now at our initial Technical Check stage, where we look for adherence to the journal's submission guidelines, including any relevant editorial and publishing policies. If there are any points that need to be addressed prior to progressing we will send you a detailed email. Otherwise, your manuscript will proceed into peer review.

You can check on the status of your submission at any time by using the link below and logging in with the account you created for this submission:

https://researcher.nature.com/your-submissions?utm_source=submissions&utm_medium=email&utm_campaign=confirmation-email&journal_id=10922

Kind regards,

Editorial Assistant
Journal of Network and Systems Management

Journal of Network and Systems Management is a transformative journal. This means it offers a hybrid publication model. When the journal accepts research for publication, the article may be published using either immediate gold open access or the subscription publishing route. For further information please visit https://www.springernature.com/gp/open-research/transformative-journals

List of Figures

List of Tables

List of Abbreviations

- AHA - Artificial Hippocampal Algorithm
- AR - augmented reality
- CNN - convolutional neural network
- DDoS- Distributed Denial-of-Service
- DoS- Denial-of-Service
- EXAI - Explainable Artificial Intelligence
- FL - Federated Learning
- ICAP - Internet Control Message Protocol
- ICT-Information and communication technology
- IIOT- Industrial Internet of things
- IOT - Internet of Things
- IP - Internet protocol
- MAC- Media Access Control Address
- PNG - Portable Network Graphics
- SDN- Software-Defined Networking
- TCP - Transmission Control Protocol
- UDP-User Datagram Protocol
- VM -Virtual Machine

Chapter 1: Introduction

SDN has emerged as a revolutionary technological breakthrough, with the capacity to alleviate the inherent constraints of traditional networks. The technique involves segregating the control plane from the data plane, enabling network administrators to centrally manage and optimize the network. The separation of the control and data planes leads to improved flexibility, scalability, and efficiency in network maintenance and management. Additionally, it provides enhanced network security by allowing administrators to implement meticulous access control policies, thereby preventing unauthorized network access. The confluence of cutting-edge Information and Communication Technologies has had a transformative impact on our society and economy, evidencing how technical progress can significantly impact a nation's Gross Domestic Product (GDP). However, the progress of technology has also amplified the probability of diverse cyber threats and malevolent attacks, which can expose even the most secure systems to vulnerabilities. In addition, advanced ICT tools, while providing several advantages, also render network security susceptible to novel vulnerabilities. The fact that cyberspace has become a sphere of operations in the field of warfare, and the US Cyber Command has been elevated to the rank of uniped combatant command, establishes the critical need to uphold the security of online data storage. Therefore, devising a comprehensive strategy that ensures the integrity of work processes and effectively neutralizes hostile cyber-attacks is of paramount importance. Finding an adaptable, scalable, and cost-effective solution to counter cybersecurity vulnerabilities and threats has become imperative, leading cybersecurity experts and academics worldwide to unite in their efforts to develop a trustworthy and secure online ecosystem.

The alarming surge in cybercrime has compelled governments and organizations to join forces to mitigate its pernicious impact. According to the Cisco Report of 2019, thirty percent of global companies fell victim to cybercrime. In recent times, the most prevalent forms of virtual outbreaks are Denial of Service (DoS) and DDoS attacks. In the case of DDoS, the attackers manipulate a cluster of compromised machines (botnet) to overwhelm their targets with requests for service termination, leading to an excessive burden on the victims and the inability to attend to genuine customers' needs. The next-generation architecture of SDN treats the control and data planes as separate entities. All management tasks, including data routing and traffic monitoring, are centralized and carried out by a software-based controller. SDNs are virtual networks that operate autonomously, delivering exceptional levels of efficiency, adaptability, and reliability. SDN enables organizations to manage their core functions securely while protecting personal user data, promoting business collaboration and data exchange with internal and external clients. The application of SDN in managing Industrial Internet of Things (IoT) devices represents a significant leap forward in the ongoing drive to enhance enterprise operations. By leveraging SDN, IoT devices can now transmit and receive data in a more secure environment that features data encryption and quick data analysis capabilities. This is a critical capability as IoT devices engender gigantic expanses of facts that require resourceful analysis to extract the vital information that can help businesses improve their operations. SDN's management of IoT devices is cost-effective, which has the potential to enhance the efficiency of applications and analytics. The integration of SDN in the IoT architecture enables more efficient and effective control of the entire IoT ecosystem, resulting in improved management, security, and data analysis capabilities. Moreover, the unprecedented growth of IoT devices and the inherent unreliability of the public Internet has necessitated the transfer of IoT traffic to private channels as soon as technologically viable. This is because the public internet is susceptible to a extensive range of cyber- spells, which can result in devastating consequences. By contrast, private channels offer greater security and reliability. In addition, delays in transferring IoT traffic to private channels could result in latency issues that may critically impact communication services and applications.

1.1Diagram of SDN DDOS ATTACK

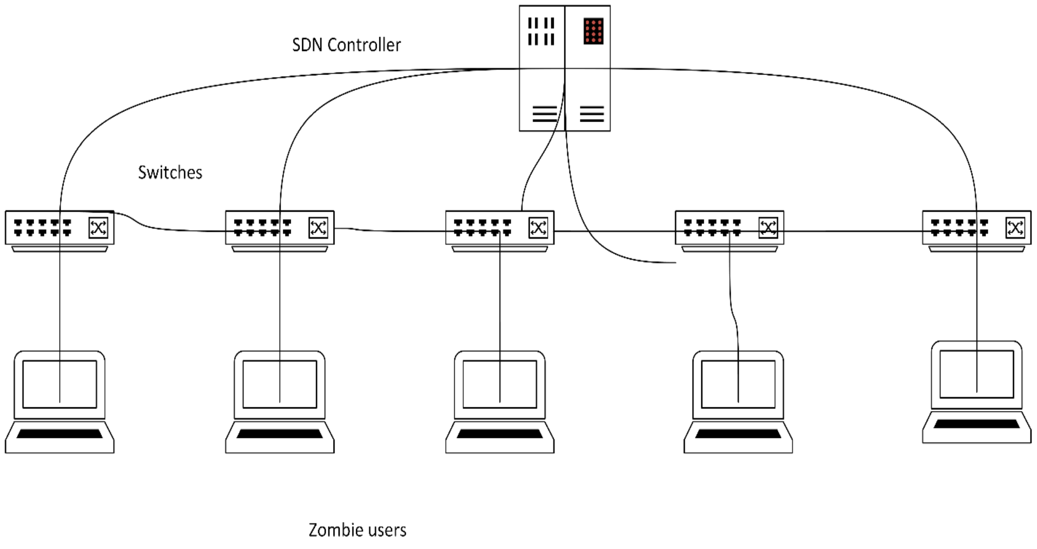


Figure 1 Diagram of SDN DDOS ATTACK

Chapter 2:LITERATURE REVIEW

This discussion delves into the intricate framework and architecture proposed for integrating convolutional neural network (CNN) ensembles into SDNs. SDNs are an emerging networking paradigm that allows for the separation of control and forwarding logic via programmability and centralized control intelligence. Commercial SDN controllers, such as Floodlight and ONOS, offer numerous networking features and components due to their scalable and cost-effective control plane. Our deep CNN ensemble architecture can be effortlessly incorporated into any off-the-shelf SDN controller, as it is centrally controlled at the control plane.

The SDN architecture comprises three distinct layers: the function level, hegemony level, and data level, each with its associated southbound and northbound APIs. To ensure digital security, cybersecurity concerns must be addressed from multiple angles, employing original and innovative research. A new framework for a deep CNN ensemble has been developed to tackle the challenge of effectively and efficiently detecting complex DDoS attacks in SDNs. Our proposed approach was evaluated using both traditional and hybrid novel bottomless absorbing techniques, demonstrating improved detection rates and reduced computational complexity.[1]

A groundbreaking technique for defending against DDoS attacks was devised using two distinctive sets of DDoS traffic data. This dataset contains benign and malicious internet activity and has a primary dataset size of 21 terabytes (TB), making it a dependable resource for various network security studies, including DDoS modeling, detection, and mitigation. While the packet sizes in malicious attack traffic range from 46 to 1,500 bytes, the benign traffic packet sizes range from 40 to 1,474 bytes. Though these characteristics are not vastly different from one another, detailed data analysis revealed various anomalies that differentiate the assault and ordinary trade. The jiffy fixed of trade smatterings was generated using the Bonesi wire-based testbed tool to simulate botnet activity. These traces were transmitted throughout the network, with botnets dispersed over the testbed's 31 subnets, each representing a separate business network. Up to 31 distinct public IP addresses can be produced by businesses using NAT, each assigned to traffic leaving private networks and entering the Internet. The traffic was created in 220 seconds, with 20 seconds of "regular" traffic and 200 seconds of "attack" traffic (TCP and ICMP), and this supplementary dataset was worn to further succession and validate the ML algorithms. A flow in this research pertains to a TCP session with four unique parameters for the TCP-SYN handshake. Results from the CAIDA 2007 dataset revealed that the network operated within its type constraints in the first 20 seconds, with low traffic volume at that time. However, the total number of flows increased significantly around the 20th second, as the total number of IPv4 addresses from which traffic originated increased, along with the number of actively used ports. It is crucial to note that ports connected to each IP address often experience a strong proliferation in the digit of open connections in the assault rung. Apiece foundation IP of the attack unbolts various ports in a short period formerly inaugural a large quantity of ports and sending many packets to the target. Nearly a thousand unique Internet Protocol addresses (IP addresses) were identified as the origins of malicious attack flows. Nevertheless, the timing of each malicious assault is unique, with more than 240 times as many source IPs as ports beginning the top assault phase. The available ports are insufficient for the vast number of possible IP addresses from which data can be sent. Only a few source IP addresses were used in the first 15 minutes of an ICMP flood attack. The network is flooded with ICMP packets when the number of ICMP packets and IPs exceeds a thousand at the 80th second and beyond, which begins to occur at the 60th second. The average pace of ICMP packet transmission for a given IP address has increased to about 860 packets every 160 milliseconds since then. The data sets suggest that an attacker can generate a significant number of TCP flows during a TCP-SYN flood without resorting to a substantial number of forged IP addresses by first opening multiple ports per one IP address.[2]

The energy consumption patterns in PNG display a gradual rate of change, as evidenced by the smooth energy consumption curves. The consistent energy usage in PNG is a testament to this fact. Research has confirmed that the suggested archetypal is above operative in terms of energy usage when the value of "+" is set to 0.2. When comparing the current model's AHA and SHG performance in terms of energy consumption, energy savings of 1J and 2J can be achieved, respectively. However, as the number of honeypots and pseudo honeypots increases, so does the power consumption in PNG. In light of this, adjustments must be made to enable the more reasonable deployment of pseudo honeypots. In cases like this, there may be a potential waste of resources that can be avoided. By implementing the pseudo honeypot technique, our suggested approach can enhance cyber resource usage while simultaneously identifying intrusions. The honeypot technique's adaptability makes this feasible. To pinpoint weak spots in the defense, they propose a counter-honeypot assault. This will enable hackers to bypass honeypots and gain access to the SDN infrastructure. The Advanced Heart Attack course has two main segments. In the first stage, the attacking force must search for honeypots in the defense system. In the second phase, they must identify the many honeypots utilized by the security system. There are various types of honeypots with different levels of interaction. Some honeypots allow for deep contact, while others only encourage surface-level interaction. The interaction that occurs with the attacker not only provides them with information about the honeypot, but also about the effectiveness of the decoys set up by the security system. Low-interaction honeypots are simpler to implement and less likely to be uncovered than other types of honeypots. However, they can only teach limited information about attackers as it is difficult for them to carry out effective assaults within the honeypot. Interactive honeypots, on the other hand, can gather more details about attackers. Nevertheless, high-interaction honeypots increase the complexity of their deployment and the danger of their upkeep by allowing attackers to engage in a wider variety of behaviors within the honeypot. Moreover, an attacker should not assume the same success rate for attacks on various honeypots, given the diverse selection of honeypots available in the market today.[3]

AR Defense is an innovative solution that offers a robust defense against the adverse effects of a DDoS attack, utilizing a straightforward and reliable algorithm. Our approach is designed to be effective even when users are being targeted from multiple directions. The superior design of our method involves conducting un-authorizations for malevolent acts at the doorway, where sachets are sent to other attendants or handlers before their dispersion. To test the success of our tactic, we simulated a DDoS attack at the application layer using a code that imitates the traffic spike that occurs during an actual DDoS attack. Once the algorithm detects an emergency, it immediately activates to protect both users and servers. The architecture of AR Defense is based on Network Function Virtualization (NFV), which employs VMs to reroute data to the appropriate server or node, guaranteeing a seamless user experience when using their telecom services.

Every user is assigned an IP address and MAC address for communicating with the network's hardware. The MAC address can be concealed from other users if necessary. Therefore, the IP address of the user or the IP address of the security real-time proxy connecting to the server is the only determining factor for whether or not an attack is launched against that user. AR Defense's impact on individual users and websites or web servers is similar to that of a DDoS attack. During a DDoS attack, the targeted node experiences a significant surge in network traffic. To prevent this, the proposed approach continuously monitors all user and server network activity. If the data flow rate exceeds the threshold, packets of information destined for the target server are instead sent to a third, specialized virtual machine (VM) that performs analysis to prevent further intrusions. For added security, all accounts belonging to individuals associated with the "server" are erased, and their IP addresses are spoofed. We used the traditional technique of writing IP addresses, as IP addresses differ by region. It is important to note that only IPv4 addresses are considered in this approach. After that, these customers are connected to a different "server."To better understand the method, let's take an example with 20,000 users on the system and fifty servers, such as virtual security system servers, proxy servers, etc. Assuming each server has a 100-unit cap, the maximum number of concurrent connections that may be made to the server at any one time is as follows. Suppose server eight is the target, and 4,000 users are connected to it. In that case, the attacker will spoof their IP addresses and assign each of them a new one.[4]

UDP is the most used transport protocol for DNS inquiries, however, it does not allow for the source IP address of DNS requests to be confirmed. This might lead to a flood of bogus DNS searches and a deluge of bloated DNS responses throughout the network. wisdom SDN was inspired by the tactics of moving target defense. These methods prevent Distributed Denial of Service (DDoS) assaults from spreading to their intended targets by keeping the attack confined to the location of the original request. There were a few overarching objectives that we kept in mind when designing Wisdom SDN. At the outset, Wisdom SDN must provide full com protection against

DNS amplification attacks. As opposed to older systems, which would first attempt to assess the network's health before trying to identify an attack, this newer technology can detect attacks immediately. Wisdom SDN takes a proactive posture against this problem by working to maintain the need for a 1 to 1 correspondence among DNS queries and DNS rejoinders cannot be understated, as it is crucial to avoid the rerouting of malevolent DNS traffic to the intended victim. To achieve this objective, Packet Authentication System (PAS) is employed. Wisdom Software-Defined Networking (SDN) employs machine learning techniques to secure the Ternary Content Addressable Memory (TCAM) OF switches, which have limited storage capacity. The primary aim of the DDoS detection module is to identify suspicious DNS requests in real-time. To effectively counter an attack, a Detection and Mitigation (DM) mechanism is employed. The system is designed to be highly scalable, while ensuring the efficacy of the approach.[5]

The article discusses the evaluation of a system's effectiveness in detecting slow DDoS attacks by calculating various metrics such as TPR, TN, FP, FN, Accuracy, Precision, and Specificity. The F1 Score, a composite metric that balances Precision and Recall, is suggested for measuring the system's performance. The system uses a SDN with flows to route traffic, and a CNNLSTM hybrid model is employed to detect low-velocity DDoS activity. The detector module assigns labels of attack traffic or benign traffic to entries based on the output of the feature extractor module, making it a binary classifier. The article also presents a diagram illustrating the use of a hybrid CNNLSTM model in the detector module.[6]

Devising effective security measures to protect against the various types of attacks that can compromise the network. The security threats posed by an SDN network can be categorized into three groups: attacks on the solicitation layer, attacks on the rheostat layer, and attacks on the data layer. However, it is the attacks on the control plane that are considered the most hazardous and tempting among the three, as this plane is liable for the smooth functioning of the network as a whole. The rheostat plane can be targeted through DDoS attacks, in which an attacker can exhaust the controller's resources by submitting a number of requests for new flow rules to the network. The impact of such an attack can be intensified by increasing the number of attackers. In addition, SDWSNs are vulnerable to other forms of assaults due to the constraints imposed by available resources. For example, because of space limitations, flow tables and buffers in SDWSN forwarding devices must be kept small, which makes them susceptible to saturation attacks. Furthermore, SDWSN networks typically have limited bandwidth and computational capability, which indicates that a saturation attack can lead to a DoS attack. Another vulnerability in SDWSNs is the absence of a gateway that links the SDN controller to the WSN. Although the controller may have sufficient resources to withstand an attack, the low-bandwidth radio module of the gateway leaves the network vulnerable to intrusion. Security approaches considered for regular SDN networks need to be modified or reshaped to address the challenges mentioned above.[7]

The veil coat, which includes the record and veil server, is the topmost coat in this DDoS protection concept and is described below as the design framework. Fog nodes and IoT devices both belong to the application layer, which sits above the cloud infrastructure. In between the veil server and the app, the layer is where we find the intermediate fog layer. The proposed effort would include malicious and no malicious machines sending cloud-based requests. In this situation, data transmission must first break through the fog before it can reach the cloud. This layer of infrastructure consists of several different gadgets, including the fog server. The SDN controller is located on the fog server, also known as the point of presence. When the nodes in the network come together, these rules will determine how they send and receive data packets For, In reality, it requires classification capabilities to evaluate whether or not incoming packets are legitimate. Each of the three algorithms described herein provides a detailed illustration of the workflow of the proposed model concerning the proposed scheme. To do this, Algorithm 1 performs a comprehensive analysis of each arriving packet C I j]. It employs not one but two separate kinds of lists: List-ND and ListL-ND. For a certain time, interval, the list designated by "Liste" contains all incoming data packets, whereas the list designated by "List" is, well, empty. As long as the "ListL" territory is not null on each data packet, the trained system will work correctly.[8]

To avoid the disruption caused by a dynamic distributed denial of service assault during the migration, an "Intrusion Detector" may be used in conjunction with an authentication server to provide both intrusion detection and intruder identification.

The SDN controller initiates the false reality for the reactive approach by sending fake traffic from other dummies VMs through the fake traffic channel on the targeted VM. The modules "Attack Profiler" and "Network Bandwidth Analyzer" provide the "Reactive Migration and False Reality Negotiator" with the crucial data needed to compute the appropriate volume of sham traffic to give the impression that the attack is still successful. That way, the attacker may keep up the pretense that the assault is successful. It is also the job of the "Reactive Migrant and False Reality Negotiator" to weigh the pros and downsides of fabricating a reality. In this way, we can be sure that keeping up appearances will be worth the effort. distributed optimization as the cornerstone of a systematic, market-driven approach to peak efficiency. Our market-driven model employs the economics of virtual markets to make the most effective use of available resources while in transit. This is done to reduce the risk of cyberattacks while maximizing the value of the scattered resources. The CSPs function as providers, aiming to provide their customers with the greatest tools at the most affordable prices. Our strategy also proposes a reputation system for virtual machines that awards or deducts points over a longer period based on the VM's past performance or failure in fending off DDoS attack threats. After a suitable migration, the site has been chosen and the target application has been migrated, an SDN controller may instruct OpenFlow to move all of the application's users to the destination virtual machine.[9]

The acquired feature data is operated as an effort for the DDoS uncovering component that is now operating in each local controller. To detect DDoS assaults, previously accomplished ML archetypal. The accomplished ML archetypal will provide the local regulator with the essential information to decide whether the movement is a DDoS assault or not. Training data must be provided to ML models. Unsupervised learning utilizes unlabeled data for training purposes. We evaluated a variety of machine learning (ML) models before opting for ELM-based models due to the efficiency of ELM training. ELM uses a random selection of the initial parameters, and the use of basic matrix operations helps to reduce the training time necessary. In contrast, several learning algorithms depend on gradient-based learning, which is a somewhat slow process. As a result, ELM is a more viable candidate for Internet of Things (IoT) real-time applications since retraining can be conducted fast and without impacting application functionality. The ELM variants for the supervised, unsupervised, and semi-supervised categories are distinct. The ELM was ruled out because of the large quantity of bandwidth that billions of IoT devices would generate. The classification of all traffic as either ordinary or DDoS is a tedious task. We also opted against using unsupervised ELM because there are billions of IoT, and the probability of routine traffic being mistaken as a DDoS assault is rather high. Therefore, we used semi-supervised ELM, a model that represents a compromise between the two groups.[10]

Because the legitimate user is blocked from using the network's system resources, the network's performance suffers as a consequence of the DoS attack. Volumetric assaults synchronized flooding attacks, fragmentation attacks, TCP state depletion assault's function layer assaults, and Flashing assaults all subtypes of DoS attacks. To utilize all network's available bandwidth, a volumetric assault will repeatedly send out ICMP echo requests or ICMP echo answers to packets. This is done so that only legitimate users of the network may access its resources. The coordination of multiple SYN requests to the server constitutes flooding. Therefore, the server will be overburdened with SYN requests, which will negatively impact system performance. To keep the server busy, the attackers in a fragmentation attack send broken packets to reconstruct. Repeatedly establishing and severing a TCP connection, known as a TCP state exhaustion attack, may cause the stable tables to become overloaded. The attacker in an application layer attack exploits security holes in the program. To do this, several application requests are sent to the server at once, keeping it busy processing these requests and so preventing it from performing other tasks. However, the phishing assault does irreparable harm to the hardware by fooling it into installing phony updates, which in turn corrupts the system and makes it unusable. As a result, the hardware becomes entirely useless and must be replaced. An integral component of the controller's job is to compile information from the switch tables. The controller is in charge of tracking the activity level of the packets in motion and cutting off the flow of those that have been inactive for a certain length of time. Examining this controller property with other data, this study aims to ascertain the effect that DoS has on packet flow across an SDN network. Newly incoming packets are collected and organized into windows of size fifty once the destination IP address has been appended. This is done because the expected number of hosts on the network may be calculated in advance. Thus, we consider all windows, calculate their entropy, and then compare it to some kind of threshold. This research was motivated by the hypothesis that a drop in entropy below a certain threshold would indicate an attack was taking place somewhere inside the network. If you want accurate results, you should either make the window as small as feasible or make it about the same size as many hosts in the network. Specifically, we evaluate the hypothesis that a window size of fifty may be used to regulate the number of new connections that can be established to each host in a network. Therefore, DDoS detection operates on the premise that, upon request, the controller will allow access to each new packet so that the database may be updated. Also, keeping the window size at 50 speeds up calculations. Additionally, it is easier and faster to identify assaults that happen within a window size of fifty packets. Using three different window widths, we can quantify not only how much CPU time the network consumes, but also how much variation there is in the entropy itself. If an attack is being conducted against the controller, it may be sensed by checking the IP address of the sachets final destinations. To address this, the controller now has a new function to create and manage a hash table of incoming packets. The number will increase by one for each new IP address that is added after this. When all fifty packets in the window have been processed, the randomness of the space will be premeditated.[11]

The act of attempting to disrupt a service by transmitting customized packets that compel the target SDN control to create a novel flow. Involves deploying a large botnet to launch the attack, typically consisting of 1500 to 4000 bots. The switch is triggered to create a forwarding rule by unique packets sent by each bot, which leads to the exhaustion of the switch's TCAM, resulting in the inability to accept legitimate traffic. Furthermore, the botnet broadcasts new packets periodically, which triggers the reactivation of installed rules, ensuring that they are not deleted due to rule timeouts. Our investigations reveal that Slow-TCAM attacks may be conducted using an extremely low quantity of interchange, up to four sachets per sec, compared to Saturation attacks, which create over a thousand packets per second, suggesting that it can bypass security measures intended to prevent high-velocity attacks. While Slow-TCAM attacks have no impact on previously installed rules, the Slow Saturation attack forces their respective timeouts to expire, resulting in their deletion. In contrast to the second version, which generates bursts of activity, the first version of the Slow Saturation attack produces a continuous stream of traffic. Our study also examines the broader impact of Slow Saturation attacks on the SDN network, beyond the affected switch, as it stresses the network's controller, causing other linked devices to behave abnormally. To our knowledge, Selective Defense is the only viable countermeasure to TCAM attacks, and the Slow-TCAM approach is still a relatively new concept (SIFT).[12]

Traditional methods for countering distributed denial-of-service (DDoS) attacks were developed for smaller, isolated networks and primarily focused on stopping volumetric attacks, the most basic type of DDoS attack. However, these methods lack autonomy and are difficult to deploy and maintain across large, geographically dispersed networks. The advent of software-defined networking (SDN) provides a new

The SDN is a novel networking paradigm that enhances traffic engineering efficiency and facilitates dynamic reconfigurations. By decoupling the data plane from the control plane through the OpenFlow protocol, the SDN concentrates all network intelligence in a single module, providing a configurable interface for effective networking. Although both traditional and SDN networks can monitor data like packet ratio, some traffic elements are hard to obtain using OF queries on the data plane. Thus, SDN-enabled systems require new approaches and tactics for detecting DDoS and botnet attacks. Although SDN is ideal for high-bandwidth applications, it is likely to become the primary vector for botnet distribution. Several research studies have concentrated on using SDN and NFV for network security, making it possible to detect and mitigate DDoS attacks earlier. However, developing DDoS attack prevention applications on top of the SDN control plane presents challenges in achieving adequate defensive performance. Sophisticated DDoS attack detection techniques are required, which may result in southbound overhead and detection lag. Despite this, the collaborative intelligence of SDN remains mostly unexplored due to the need for precise detection and quick response to prevent controller saturation.[13]

SDN is a single controller responsible for configuring and managing all devices that forward data across the network. The controller is responsible for supervisory governing all network features. OpenFlow switches are responsible for delivering packets of data to their final destinations, with flow entries in the flow table enabling their operation. A rule, an action, and counters are the three components of flow entries. A rule may utilize multiple header fields as potential matching criteria. The packet flow regulations are already established by the controller. The action field enables the switch to be tailored to the user's requirements. If header fields don't comply with the flow table requirements, the switch will send a packet to the controller, which will either reject or accept the packet based on the analysis result. The SDN controller communicates with the switch via the low-mod rule to update the flow table. In a DDoS attack, the attacker tries to exhaust the server's resources as quickly as possible. When an SDN is attacked, an influx of new packets is delivered to the SDN routers. In this example, the host acts as an attacker by bombarding the SDN switch with packets. As a result of the controller receiving several messages that do not satisfy the flow table rules, the controller sends out multiple Flows Mod messages to update the flow table in the switch. The controller will keep broadcasting these Flow Mod rules until it reaches its limits, depleting its resources and causing network congestion due to the sheer volume of packets. The SDN architecture is vulnerable to various DDoS attacks, two of which are shown in the example: congestion on the controller-switch channel and controller fatigue. The processing power of the SDN network is reduced as a result of these attacks.[14]

SDNs utilize two main types of controllers: open-source and proprietary. Open-source controllers differ in their programming languages, with examples including Floodlight, Ryu, Open Daylight, Open Contrail, and ONOS. Open Flow is a common SDN controller, and Open Daylight is a popular option used by numerous companies and developers worldwide. ONOS, which utilizes Java-based code deployed within an Open Service Gateway Initiative (OSGi) container, is highly scalable and fault-tolerant, making it well-suited for large organizations and service providers. Commercial, closed-source controllers are typically distributed to improve their failure tolerance, with examples including the Cisco Open SDN Controller, HP VAN Controller, and VMware NSX. The Cisco Open SDN Controller, based on the Open Daylight controller, offers features such as clustering, serviceability, and open virtual appliance (OVA) file packaging, and supports both Java APIs and northbound REST APIs. The HP Virtual Application Network Service Delivery Controller simplifies management, provisioning, and orchestration through a unified interface, while the VMware NSX SDN controller offers benefits such as improved workload mobility and reduced network provisioning time, thanks to its hypervisor platform for network virtualization. Overall, both open-source and proprietary SDN controllers offer unique advantages and may be well-suited for different types of organizations and applications.[15]

A tuple space can be likened to a persistent memory storage that can temporarily contain tuples that will be subsequently retrieved by a process. The originator of a tuple is commonly referred to as the producer, while the process that reads tuples from the storage is called the consumer. The consumer and producer roles may be executed by a single process. The communication between the two roles is separated spatially, enabling any process to store a tuple that is available for other processes to read. This is made possible by the fact that the process that generated the tuple does not need to be known for other processes to access it. This temporal independence allows any process to access the tuple at any given time. The functionality of tuple spaces has the potential to be useful in the design of SDN, specifically in the temporary storage of incoming packets by switches and the subsequent retrieval of appropriate flow rules defined by controllers. Tuple spaces are similar to shared memories in that several processes can access the data contained within them. However, tuple spaces can be implemented independently of their logical comprehension. If implemented correctly, they could function as central storage hubs. However, doing so would increase the likelihood of a single point of failure. Modern tuple spaces are developed as distributed middleware, with a copy of the space stored locally on each node in the network. This configuration offers the advantage of not requiring any extra hardware, as it can be efficiently distributed over all the switches and controllers. The local process accesses can be audited independently using various middleware components, making it easier to track the number of times switches visit the tuple space. This can provide insight into whether a flooding attack is taking place.[16]

In order to counter DDoS attacks, a new approach has been developed that utilizes the power of SDN and NFV. This approach leverages a novel technique for distinguishing between legitimate traffic and an attack and allows for the rapid transition to Pushback mode to halt the threat and prevent network congestion. The SDN controller responsible for the network can monitor traffic levels on switches to detect an attack, create a filtering rule, and activate the Pushback mechanism. Firewalls may be established as VNFs to prevent attack traffic from entering the SDN domain. If multiple SDN domains are interconnected, controllers can collaborate to eliminate harmful flows. This approach advances the state-of-the-art Pushback mechanism and enables networks to effectively safeguard against DDoS attacks.[17]

Using all of a network's IP addresses effectively. It has been noticed that IPv4 addresses are not depleted when a subnet uses them. An IP packet has to be forwarded, the controller chooses an IP address from the pool and assigns it to the packet. Next, the controller updates the packet's header by changing the destination address field to reflect the new choice. In our architecture, SDN routers are only deployed at the border routers responsible for regulating incoming and outgoing traffic. The Topology Manager: The Topology Manager module analyses the presently reachable nodes and the connections between them to establish the kind of topology. In this piece, we use the mesh topology, which may be either a whole or a broken mesh. To establish the topology, it will first broadcast a request message to all nodes. When a node receives a request message, it will respond with a message that details its proximity to other nodes, the cost of the connection, the operational mode of the link (simplex, duplex, etc.), the propagation delay, and so on. This data is included in the reply packet and sent to the controller as soon as it is available.[18]

The distribution abstraction will transform the distributed control problem into a logically centralized one, protecting SDN applications from the inconsistencies that arise with dispersed states. In SDN terminology, the Network Operating System (NOS) serves as the central distribution layer required for its execution. There are two critical functions that this layer provides. Its primary function is to instruct the devices used for forwarding how to behave. Second, it collects forwarding layer status information, such as the devices and connections in the network, to provide network applications with a complete image of the network. Specifications are the highest degree of abstraction, and they allow a network application to describe the desired behavior of the network without having to take on the burden of actually implementing it. Several virtualization methods and network programming languages may assist in achieving this goal.[19]

In order to determine the best route between two hosts in a network, the QRS takes into account various factors, including jitter, packet loss, and link utilization. To assist in this process, the SDN controller has a comprehensive view of the network architecture and can gather high-quality service data from OpenFlow switches. To monitor connection latency, which is currently costly and complex, the LLDP protocol is used, a vendor-neutral layer two protocol that can measure connection latency with millisecond-level accuracy and minimal overhead. OpenFlow's native packets, such as LLDP, Packet-Out, Packet-In, and Echo messages, are used to reduce overall network traffic and conduct more precise real-time monitoring. The LLDP packet is modified to include time stamp data in a TLV field. Echo messages and link discovery methods.[20]

2.2 LITERATURE Analysis Table 1

Citation.no	Decree
1	The convolutional neural network functions effectively inside the CNN ensemble architecture.
2	TCP Flooding attack and Internet Control Message Protocol (ICMP) flood assault in SDN-based ISP networks.
3	When using a double honeypot, detection performance is prioritized.
4	Different protocols, such as UDP and TCP, may have different thresholds.
5	Wisdom SDN has reached the level of complete intelligence for DDoS.
6	Through the use of the SDN controller's REST API, the detection module can gather traffic flow statistics from SDN switches and then analyze this data to identify a slow DDoS assault.
7	There is a need to increase the performance of solutions, and recent approaches in the literature for identifying DDoS assaults in SDWSN do not necessarily take limited networks into account.
8	The traffic resulting from DDoS attacks is analyzed and scrutinized.
9	frequency of migration optimized to reduce network resource waste while protecting against attacks
10	Superior to state-of-the-art solutions in throughput by twenty-one%
11	SDN network and determine the amount of performance loss that may be attributed to a DDoS assault on an SDN network.
12	Attacks against software-defined networks that cause a slow denial of service
13	There is a need for flexible and dynamic techniques to secure and grow fog-to-things infrastructure, and the possibility for an SDN-based architecture has been suggested.
14	By dynamically managing its infrastructure and services, SDN offers a viable solution to networking consumers.
15	SDN offers several security-related characteristics.
16	SDN control plane decentralization and tuple spaces
17	Pushback is launched in the event of a massive volume assault that exceeds the capacity.

Table 1 LITERATURE Analysis

Chapter 3 : Data Set

3.1 Data set

Using the Mininet emulator to produce an SDN-specific dataset, machine learning and deep learning methods were used to improve traffic categorization. In order to simulate both legitimate (TCP, UDP, and ICMP) and malicious (TCP Syn attack, UDP Flood attack, and ICMP assault) network traffic, the initial phase included setting up a single Ryu controller to handle ten separate Mininet topologies. The dataset included 23 characteristics, including switch-id, packet and byte counts, duration, source and destination IP addresses, transmission and reception rates, which were either directly gleaned from the switches or computed. There was also a column reflecting the security of the traffic, with one being benign traffic and zero denoting hostile traffic. 1,04,345 rows of data were gathered during a 250-minute network simulation period, enabling the improvement of traffic categorization skills.[21]

3.2 Table 2 Dataset specification table

Flow monitoring interval	30 sec
Number of classes	2
Class label 0	Benign traffic
Class label 1	malicious traffic
Network simulation is run	250 minutes
Total Data collected	1,04,345

Table 2 Dataset specification table

3.3 Table 3 SDN DDOS Dataset table

Extracted Features	Calculated Features
Switch-id	Packet per flow which is the packet count during a single flow
Packet count	Byte per flow is the byte count during a single flow
byte count	The packet Rate is number of packets send per second and calculated by dividing the packet per flow by monitoring interval
duration_sec	number of Packet_ins messages
duration_nsec which is duration in nanoseconds	total flow entries in the switch
total duration is sum of duration_sec and durstaion_nsec	tx_kbps
Source IP	rx_kbps
Destination IP	data transfer
Port number	receiving rate
tx_bytes is the number of bytes transferred from the switch port	Port Bandwidth is the sum of tx_kbps and rx_kbps.
rx_bytes is the number of bytes received on the switch port	
dt field show the date and time which has been converted into number.	

Table 3 SDN DDOS Dataset table

3.4 Label encoding & standardization, data splitting.

The term "label encoding" refers to the process of translating human-readable labels into a machine-readable numerical format. This will help machine learning algorithms determine the best way to apply these labels. In supervised learning, Preparing structured datasets is an important step that ensures successful outcomes. Standardising data is a great way to transform it into a normal distribution! The standard normal distribution has a mean of 0 and a standard deviation of 1, which is great because it allows for easy calculations and comparisons in statistical analysis. It is often represented by the letter Z and is commonly used in statistical analysis to standardize normal distributions so that they can be compared and analyzed using the same criteria. The standard normal distribution is also used to calculate probabilities associated with normal distributions. Splitting a large dataset into several smaller ones is what "data splitting" refers to. When data is split in two, the first part is often used for hypothesis testing, while the second part is used for training the model. When it comes to building models that are informed by data, data splitting is an essential component of data science. This technique facilitates the construction of correct data models and the execution of operations that use data models, such as machine learning.

Chapter 4 :Methodology

4.1 XG boost

XG Boost is a popular machine learning algorithm that uses a regularized objective function to minimize the dissimilarity between predicted and target outputs, along with a penalty term for model complexity. This convex loss function is used to train regression tree functions, with L1 and L2 regularization options. During the training process, the algorithm iteratively adds additional trees are utilised to forecast the residuals or discrepancies of the preceding trees.The output of these trees is combined with the output of the prior trees to make the final prediction. The name "gradient boosting" comes from the use of a gradient descent method to reduce the model loss during multiplication. This approach results in a powerful ensemble model that can accurately predict a wide range of outputs, Due to its versatility and efficacy, it has become a widely adopted option for various implementations in the fields of data science and ML.[22]

4.2 Random forest

Random forests is a machine learning algorithm that combines the predictions of several randomized decision trees and takes the average. This technique has been successful in situations with a large number of variables and observations, and it can be easily adapted to different machine learning tasks. Random forests produce metrics of varying importance, which can be used to tackle large-scale problems. However, the theory behind this algorithm is more complex than it appears, and current research is focusing on parameter choice, resampling mechanisms, and measuring variable importance. This article provides an overview of the underlying mathematical processes that drive the algorithm and highlights the latest theoretical and methodological developments in this field.[23]

4.3 Shap

SHAP's goal is to provide light on how each characteristic contributed to the prediction of instance x. To calculate Shapley values, coalitional game theory is used in the SHAP explanation approach. Coalition members are the values stored in an instance's attributes. How the "reward" (= prediction) should be divided across the qualities is described by Shapley values. In tabular data, for instance, a player may stand in for a single feature value. A player may also be thought of as a group of feature value values. Superpixels may be used to arrange pixels and distribute the prediction to characterize an image. The linear SHAP model, upon which the Shapley value explanation is based, is an additive method for attributing features. From this vantage point, Shapley's values may be seen in LIME. Here is a breakdown of the SHAP: [24]

4.4 LIME

LIME provides locally correct explanations near the instance being discussed by sampling and collecting a proxy dataset. It creates five thousand normal distribution samples of the feature vector by default (see the num samples option). The target variable is then extracted from these five thousand samples using the prediction model whose choices it is seeking to explain. Selecting Characteristics from the Substitute Dataset: After obtaining the surrogate dataset, each row is weighted by its resemblance to the initial sample/observation. The system then retrieves the most important properties using a feature selection method, such as Lasso. In addition, LIME applies a Ridge Regression model on the samples using the retrieved values alone.[25]

4.5 Dalex

Typically, the performance of a model is summarized by a single statistic, such as F1 or accuracy. This allows models to be ranked and the best model to be chosen. However, when attempting to comprehend a model, descriptive data are desirable. ROC (Receiver Operating Characteristics) is the most often used descriptive statistic for classification, and Hernandez-Orallas describes several extensions for regression.[26]

4.6 Explainer Dashboard

Explainer Dashboard is a module for Python that, when run, creates interactive dashboards that provide users the ability to explain and grasp how the model works and what the outcome will be. "Black Box model" is the name given to a model that does not have this kind of tool. As a result, it becomes difficult to explain the reasoning behind the judgments made by the model as well as the factors that impact those conclusions. Publish a dashboard in the form of a web application in Jupiter or Co lab notebooks with the help of this Python function. It is quite easy. This web application demonstrates the operation of the applicable machine learning model by way of a series of illustrative diagrams that users may interact with. Scikit-learn, XG Boost, Cat Boost, Light GBM, and a variety of additional algorithms may be used in the construction of these models. The dashboard provides insights into the performance of the model through the use of interactive plots, such as SHAP value plots to explain feature reliance, SHAP Interaction plots for feature interactions, Partial Dependence Plots (PDP) for the influence of chosen features, and Decision Trees, amongst other visualizations.[27]– [30]

4.7 Explainable Boosting Machines

The Explainable Boosting Machine, often known as EBM, is a Generalized Additive Model that is tree-based, makes use of cyclic gradient boosting, and incorporates automatic interaction detection. EBMs often achieve the same level of precision as the most innovative black box models while also preserving their capacity for complete interpretability. Even while EBMs often need more time to train than other algorithms that are already in use, they are far more condensed and faster when it comes to producing predictions.[31]

4.8 ELI5

There are several websites, such as Yahoo Answers and Quora, in addition to a wide variety of Reddit boards and subreddits, which provide forums to debate open-ended issues. Our primary focus is on the subreddit known as Explain Like Elie Five (ELI5), in which users are encouraged to respond that a kid of that age would be able to comprehend. The solutions are intended to be completely self-contained, and as a result, they require less previously learned knowledge of the world. Furthermore, 3ELI5 makes use of simpler language that is easier to model, which adds to the sense of intrigue that surrounds the test.[32]

4.9 Federated Learning(FL)

FLis a ML methodology that can be used to train a single model on several nodes or devices without exchanging raw data. Each node in a federated learning network autonomously trains its model with its own and then shares the improved model with a master node. The server compiles and merges all-new model versions into the master, global model. Once the global model has been modified, it is sent back to the devices so they may use it to enhance their own representations. This procedure is repeated until the global model achieves an acceptable level of accuracy. The most significant benefit of federated learning is that it removes the need to centralize data, enabling machine learning models to be trained utilizing any accessible data. This allows models to be trained without compromising the confidentiality of sensitive or secret data, which is particularly useful in circumstances when the data is sensitive or confidential. The concept of FL can be visualized in the algorithm and the description of the terminology.

4.10 Federated Learning Algorithm

- Algorithm 1 – Federated average. The j client is indexed by p; G is the resident minimum batch extent E is the local quantity of eons and R is the erudition rate.
- Server implements.

```
Make ready w0 For apiece spherical t= 1,2,3,4,5,6,7,8,9
"j←max (F ·j,1)"
Ht←( haphazard set of j patrons)
For apiece haphazard j ∈Ht in parallel do
Wij+1 ←Client update( j ,Wi)
$W_{i+1}$  $W_i^j$+1
Client update (j,W): || run on client j
G←1(split Rj into Batch size of G)
For each local epoch E from 1 to 46 do
For batch g ∈G do W←w-m∇U(W;G)  Return W to the sever
```

4.11 The FL Algorithm terminology

W_i-model weights on communication round #i

W_i^j - model weights on communication round #i on client #j

A- The fraction of clients performing computations on each round is being considered.

D- The frequency with which each client undergoes training iterations on a given dataset during each round.

G- The mini-batch size utilised for modifications by clients at the local level.

m – learning rate

Q_j- set of data point on client j

R_j – Number of data point on client j

E_i(W)- Loss U (A_x,$B_{x,}$C) with parameters W

p – indexing

Equation

$Fj\ (W)\ = \frac{1}{nj} \sum_{j \in Q_j} \quad E_j(W)$ (1)

$gj\ =\ \nabla Fj\ (Wj)$

This corresponds to a full-batch (non-stochastic) gradient descent. For the current global model W^i, the average gradient on its global model is calculated for each client *J*.

F- Fraction of clients participating in that round T - No. of training passes each client makes over its local dataset each round G - Local minibatch size used for client updates.

4.12 Experimental Setup for FL Table 4

Platform	Google Collab
Capabilities	Tesla T-4 GPU 16 GB GGDR6 DISK 78.19 GB
Federated Learning	Master Slave Model
Data Set	described in section
hyperparameters	NO of rounds -9 Epoch -45 Bach size -10 Weight scaling factor Weigh federated average. Q Federated average Learning rate -10 Comms round -10 Optimizer SGD

Evaluation Matrix	Accuracy = $\frac{Number\ of\ \ corract\ perdication}{Toltal\ Perdication}$
	Precision = True Positives / (True Positives + False Positives)
	Recall = True Positives / (True Positives + False Negatives)
	F1 Score= (2 * Precision * Recall) / (Precision + Recall)

Table 4 Experimental Setup for FL

4.13 SDN DDOS FEDERATED LEARNING ALGORITHM WORKING Figure number2.

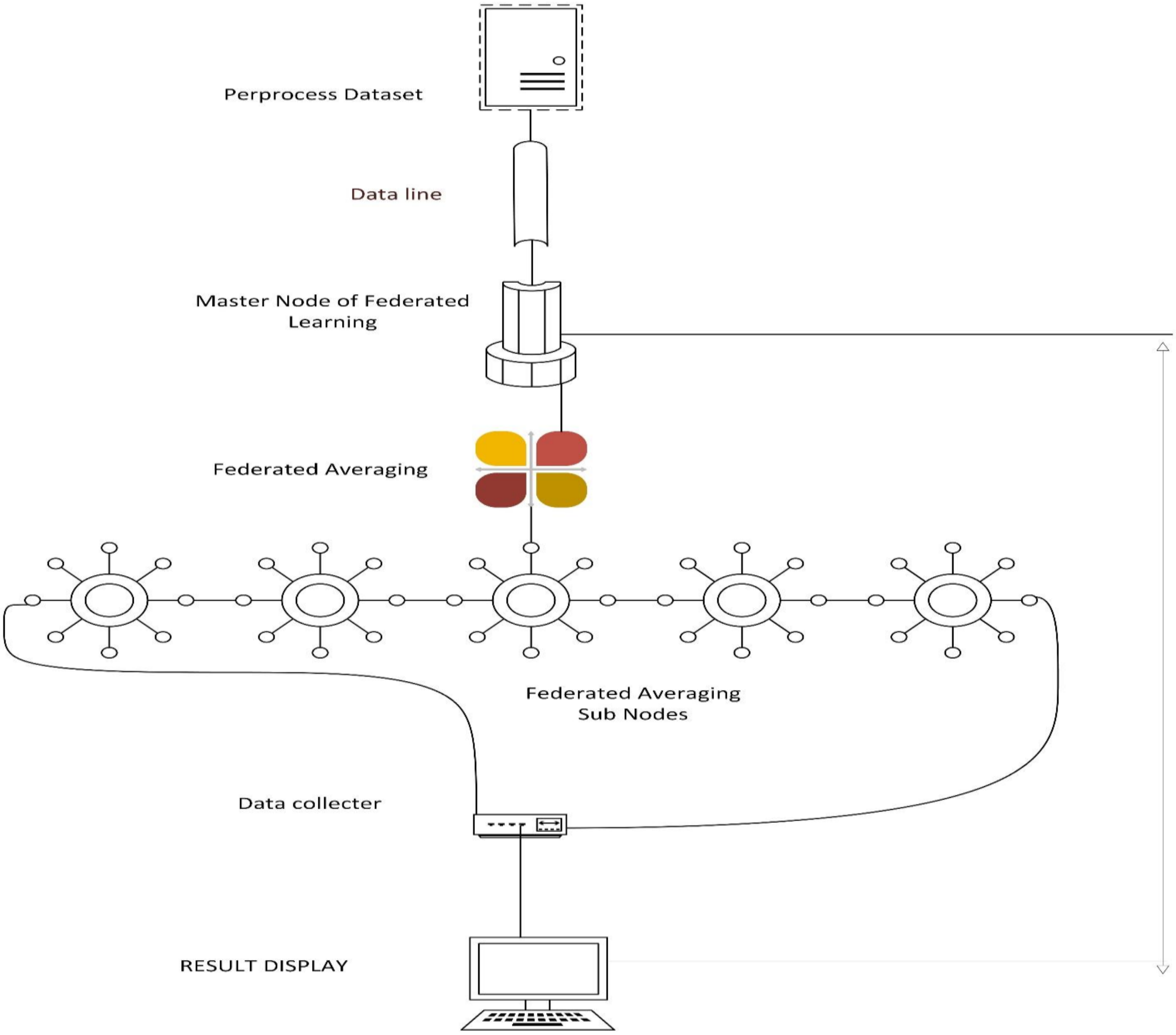


Figure 2 SDN DDOS FEDERATED LEARNING ALGORITHM WORKING

The diagram presented above illustrates various methods for mitigating DDoS attacks on the FL base. The phenomenon under consideration consists of a sequence of seven distinct stages. Each stage holds its own significance in sequential memory.

4.14 Mathematical Equation’s

a. Data Set: D = {(x1, y1), (x2, y2), ..., (xn, yn)}

b. Label Encoding : y' = encode(y)

c. Standardization: X' = standardize(X)

2. Data Splitting: Dtrain, Dtest = split(D, train_size)

a. XG Boost: G(x) = argmax sum(y - Xw)^2 + sum(g(w)), where g(w) is the regularization term

b. Random Forest: F(x) = argmax sum(y - Xw)^2 + sum(g(w)), where g(w) is the regularization term

c. Model Shape: S = (n1, n2, ..., nl), where nl is the neural network's output layer nodes.

d. LIME: LIME(x) = argmax w * f(x') + R(w), where x' is a perturbed version of x, w is the weight vector, f is the black-box model, and R is the regularization term

e. DALEX: DALEX(x) = argmax w * f(x) + R(w), where w is the weight vector, f is the black-box model, and R is the regularization term

f. Explainable Dashboard: D(x) = LIME(x) + DALEX(x)

g. Explainable Boosting Machine: EBM(X) = argmax sum(y - Xw)^2 + sum(h(w, xi)), where h is the weak model, and w is the weight vector.

h. EL5: EL5(X) = argmax sum(y - Xw)^2 + sum(h(w, xi)) + sum(l(w, xi)), where l is the local loss function, and w is the weight vector.

i. Federated Learning: FL(X) = argmax sum(y - Xw)^2 + sum(h(w, xi)) + sum(l(w, xi)), where l is the local loss function, w is the weight vector, and X is the aggregated data from multiple clients.

j. Masters salve Model: MLM(X) = argmax sum(y - Xw)^2 + sum(h(w, xi)) + sum(l(w, xi)) + sum(p(w, xi)), where p is the privacy-preserving function that ensures the privacy of clients' data.

k. Final Prediction: y' = predict(X), where y' is the predicted label for a given input X

Chapter 5: Results Discussion

5.1 Results Discussion

The present inquiry pertains to the matter of DDoS assaults in SDN through the utilisation of federated learning. This approach guarantees confidentiality during the training of a universal model, with several SDN controllers functioning as clients. The methodology utilised yielded a noteworthy precision rate of 99.39% in forecasting DDoS attacks, rendering it appropriate for prompt identification and remedial actions. The model's performance was assessed using various metrics, such as accuracy, precision, recall, and F1 score, which offer distinct viewpoints on the model's abilities. The utilisation of feature selection methods is of paramount importance in achieving precise output predictions, especially in datasets with high dimensions. This is because irrelevant features can lead to an increase in computational difficulty, preparing time, and out-of-sample accomplishments. The SHAP open-source Python package is a valuable resource for elucidating the impact of individual input parameters on predictions, with a particular emphasis on the Shapley value. The study's findings suggest that the utilisation of federated learning within SDN to identify DDoS attacks is a viable method for safeguarding privacy. Moreover, this approach has the potential to be applied to other security-related functions within SDN.

5.2 Shap results

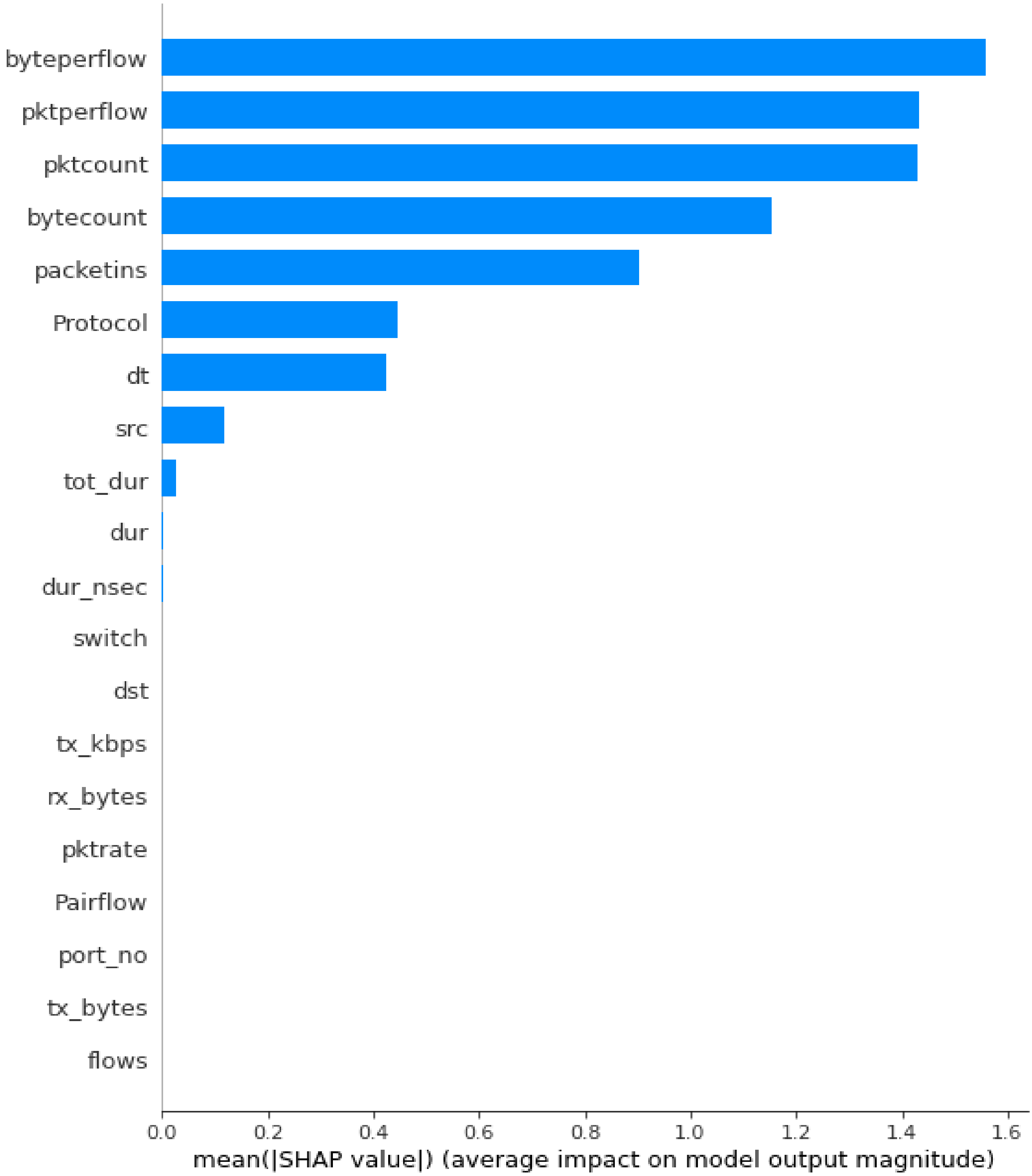


Figure 3 Shap results

The presented bar graph illustrates the SHAP values of a dataset consisting of 50 features. The utilization of SHAP values for 9 features, as depicted in the aforementioned graph, is crucial in the identification of DDoS attacks in Software-Defined Networking (SDN).

5.3 Dalex Results

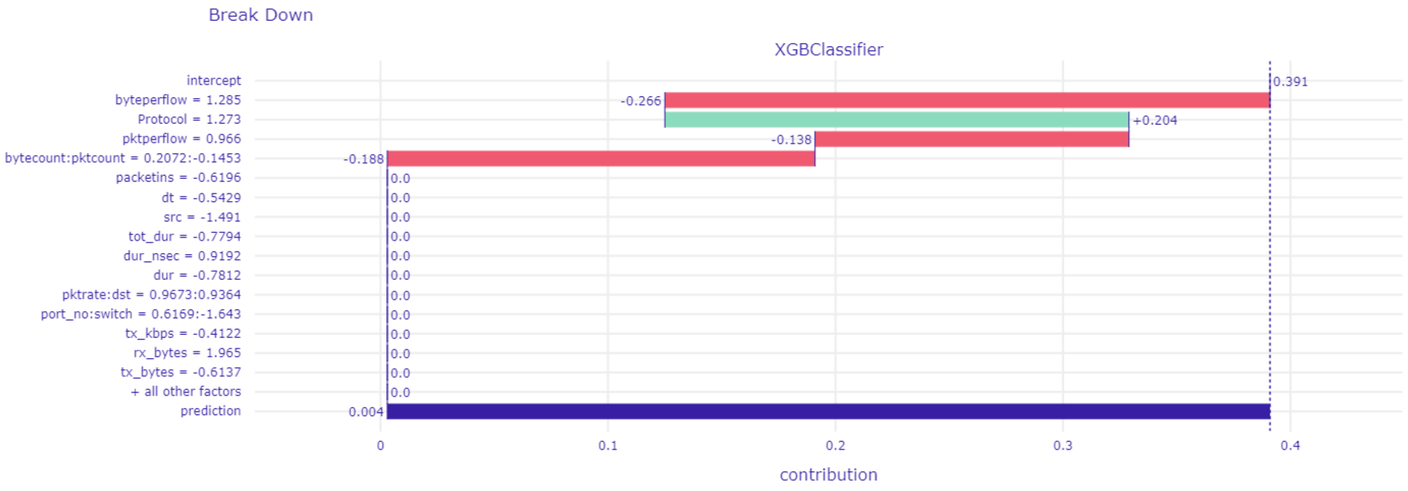


Figure 4 Dalex result break down.

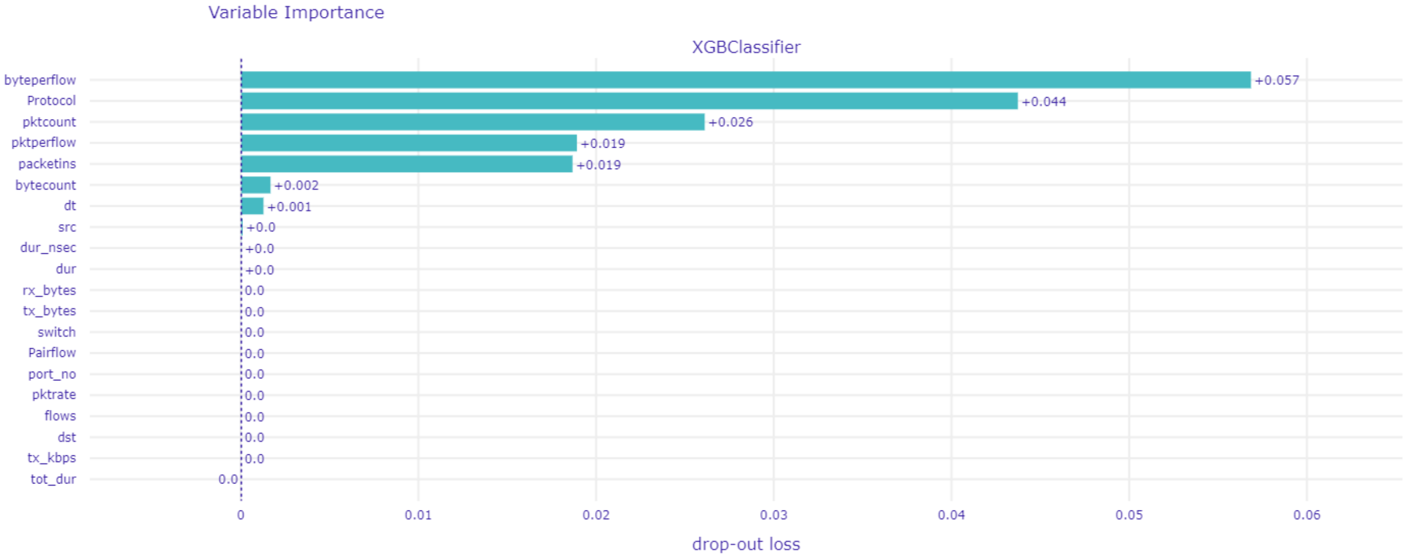


Figure 5 Dalex result Variable importance

The presented visual displays a comparative analysis of contributions from dataset features, with corresponding numerical values. The second bar graph displays the significance of variables in relation to dropout loss, utilising the XGBoost classifier.

5.4 EBM

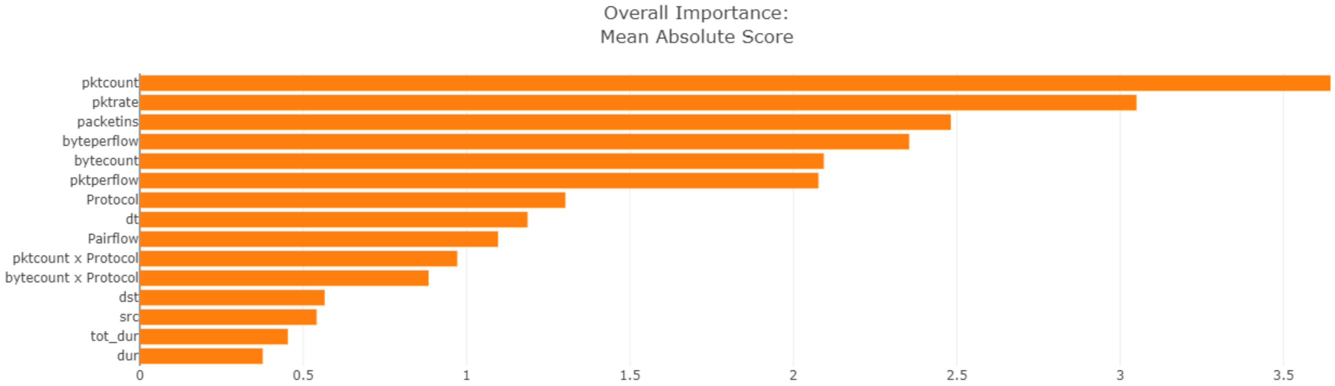
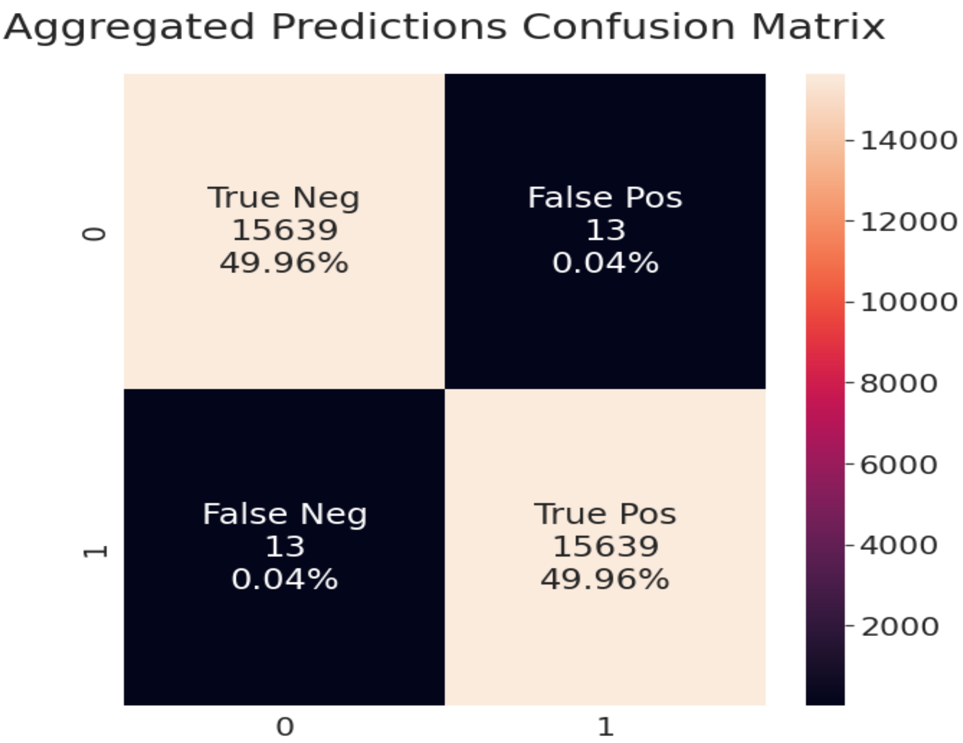


Figure 6 EBM Overall Importance Mean Absolute Score

The graph depicted above displays the significance of dataset features in its entirety, as determined by the mean absolute score.

5.5 Table 5 Experimental results of federated learning in the prediction of DDOS attack in SDN

Round	accuracy	recall	precision	F1 score
1	98.90%	98.90%	98.90%	98.90%
2	98.83%	98.83%	98.83%	98.83%
3	99.32%	99.32%	99.32%	99.32%
4	99.08%	99.07%	99.08%	99.07%
5	99.39%	99.39%	99.39%	99.39%
6	99.33%	99.33%	99.33%	99.33%
7	99.17%	99.17%	99.17%	99.17%
8	99.54%	99.54%	99.54%	99.54%
9	99.39%	99.39%	99.39%	99.39%


Aggregated Predictions Confusion Matrix
True Neg
15639
49.96%
False Pos
13
0.04%
False Neg
13
0.04%
True Pos
15639
49.96%
0
1
0
1
14000
12000
10000
8000
6000
4000
2000

Chapter 6: Conclusion

6.1 Conclusion

The matter of ensuring the security of SDN networks has emerged as an escalating apprehension in contemporary times, with DDoS attacks being one of the most noteworthy menaces. The aforementioned attacks have the potential to inflict considerable harm on network infrastructure, resulting in service disruptions for genuine users. The identification and mitigation of DDoS attacks is of paramount importance in guaranteeing the accessibility and dependability of SDN infrastructures. Fl is a potential approach for detecting DDoS attacks in SDN networks. This technique utilizes artificial intelligence to enable multiple devices to collaboratively train a model while maintaining the privacy of their data, without the need for a central server. The proposed methodology involves training a centralized model on the local data of individual clients, followed by the aggregation of updated model parameters to generate a global model. The aforementioned iterative process persists until the model attains a satisfactory degree of precision. I posited a federated learning methodology for detecting DDoS attacks in SDN networks as a part of their research. The methodology was designed with the objective of enhancing the precision and effectiveness of DDoS identification, while upholding the confidentiality of data. The study centered on two pivotal facets, namely the efficacy of the Fl methodology and its capacity to maintain data privacy. In order to assess the efficacy of the suggested methodology, experiments were carried out utilizing a publicly accessible dataset of network traffic. The dataset comprised data pertaining to packet size, protocol type, and destination IP address, which were utilised to facilitate the training of the federated learning model on the local data of each client. Subsequently, the revised parameters of the model were gathered to establish a comprehensive model that was employed for forecasting DDoS assaults. The findings of the experiment indicate that the employment of federated learning methodology yielded a notable level of precision in forecasting DDoS attacks, while simultaneously preserving the confidentiality of the data. The methodology exhibited commendable efficacy, characterized by minimal communication overhead and expeditious convergence rate. The safeguarding privacy characteristics of the recommended strategy were evaluated through the utilization of differential privacy, which involves the introduction of random fluctuations to the data in order to mitigate potential attacks.

The findings indicate that the utilization of federated learning methodology can effectively preserve the confidentiality of data while simultaneously attaining a notable level of precision in detecting DDoS attacks. The study's findings indicate that the suggested federated learning methodology for detecting DDoS attacks in SDN networks exhibited favorable outcomes with regards to precision, effectiveness, and confidentiality maintenance. The aforementioned methodology has the potential to function as a significant instrument in safeguarding SDN infrastructures from DDoS assaults. Potential future research endeavours may entail expanding the methodology to encompass alternative forms of network assaults and integrating supplementary methodologies to enhance efficacy. The utilisation of federated learning methodology in the context of SDN networks presents a promising avenue for fortifying the security of such networks against a diverse range of security threats, particularly in the realm of DDoS attack detection.

6.2 Future work

The identification of crucial network features that signify a DDoS attack in SDN networks can be achieved through the implementation of a genetic algorithm. The aforementioned task is achieved through the process of instructing the algorithm using a collection of network traffic data, where every characteristic denotes a plausible resolution to the issue of identifying DDoS attacks. Subsequently, the algorithm assesses the suitability of every characteristic and employs genetic operators to produce novel characteristics. With the passage of time, the algorithm gradually approaches a subset of characteristics that exhibit a strong predictive capability for detecting DDoS attacks. The utilisation of a genetic algorithm in the context of detecting DDoS attacks within SDN networks presents numerous benefits. Initially, it facilitates the recognition of the most significant characteristics without necessitating manual selection of features, a process that can be both time-consuming and susceptible to errors. Furthermore, the system is capable of effectively managing datasets with high dimensions, a frequent occurrence in SDN environments, through the automated reduction of features utilised in the detection procedure. Thirdly, the system has the ability to adjust to evolving network circumstances by consistently revising the collection of characteristics employed in the identification procedure. Researchers employed a genetic algorithm in a study to identify DDoS attacks within a SDN framework. The research revealed that the genetic algorithm exhibited a notable capability to effectively identify DDoS attacks, demonstrating a considerable level of precision and recall. In addition, the algorithm exhibited the capability to adjust to dynamic network circumstances, rendering it a resilient approach for identifying DDoS attacks within SDN networks.

References

[1] S. Haider *et al.*, “A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks,” *IEEE Access*, vol. 8, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.

[2] N. N. Tuan, P. H. Hung, N. D. Nghia, N. van Tho, T. van Phan, and N. H. Thanh, “A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN,” *Electronics (Switzerland)*, vol. 9, no. 3, pp. 1–19, 2020, doi: 10.3390/electronics9030413.

[3] M. Du and K. Wang, “An SDN-Enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things,” *IEEE Trans Industr Inform*, vol. 16, no. 1, pp. 648–657, 2020, doi: 10.1109/TII.2019.2917912.

[4] A. K. Singh, R. K. Jaiswal, K. Abdukodir, and A. Muthanna, “ARDefense: DDoS detection and prevention using NFV and SDN,” *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, vol. 2020-Octob, pp. 236–241, 2020, doi: 10.1109/ICUMT51630.2020.9222443.

[5] Z. Abou El Houda, L. Khoukhi, and A. Senhaji Hafid, “Bringing Intelligence to Software Defined Networks: Mitigating DDoS Attacks,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2523–2535, 2020, doi: 10.1109/TNSM.2020.3014870.

[6] B. Nugraha and R. N. Murthy, “Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks,” *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020 - Proceedings*, pp. 51–56, 2020, doi: 10.1109/NFV-SDN50289.2020.9289894.

[7] G. A. N. Segura, S. Skaperas, A. Chorti, L. Mamatas, and C. B. Margi, “Denial of service attacks detection in software-defined wireless sensor networks,” *2020 IEEE International Conference on Communications Workshops, ICC Workshops 2020 - Proceedings*, 2020, doi: 10.1109/ICCWorkshops49005.2020.9145136.

[8] R. Priyadarshini, R. Kumar Barik, and H. Dubey, “Fog-SDN: A light mitigation scheme for DDoS attack in fog computing framework,” *International Journal of Communication Systems*, vol. 33, no. 9, pp. 1–13, 2020, doi: 10.1002/dac.4389.

[9] S. Debroy *et al.*, “Frequency-Minimal Utility-Maximal Moving Target Defense against DDoS in SDN-Based Systems,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 890–903, 2020, doi: 10.1109/TNSM.2020.2978425.

[10] N. Ravi and S. M. Shalinie, “Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture,” *IEEE Internet Things J*, vol. 7, no. 4, pp. 3559–3570, 2020, doi: 10.1109/JIOT.2020.2973176.

[11] A. Dumka, A. Ashok, and P. Verma, “Performance Analysis of DDoS Attack on SDN and Proposal of Cracking Agorithm,” *International Journal of Information Technology Project Management*, vol. 11, no. 4, pp. 1–12, 2020, doi: 10.4018/IJITPM.2020100101.

[12] T. A. Pascoal, I. E. Fonseca, and V. Nigam, “Slow denial-of-service attacks on software defined networks,” *Computer Networks*, vol. 173, no. October 2019, 2020, doi: 10.1016/j.comnet.2020.107223.

[13] P. Krishnan, S. Duttagupta, and K. Achuthan, “SDN/NFV security framework for fog-to-things computing infrastructure,” *Softw Pract Exp*, vol. 50, no. 5, pp. 757–800, 2020, doi: 10.1002/spe.2761.

[14] R. Swami, M. Dave, and V. Ranga, “Voting-based intrusion detection framework for securing software-defined networks,” *Concurr Comput*, vol. 32, no. 24, pp. 1–16, 2020, doi: 10.1002/cpe.5927.

[15] A. M. Abdelrahman *et al.*, “Software-defined networking security for private data center networks and clouds: Vulnerabilities, attacks, countermeasures, and solutions,” *International Journal of Communication Systems*, vol. 34, no. 4, pp. 1–20, 2021, doi: 10.1002/dac.4706.

[16] S. Belguith, M. R. Asghar, S. Wang, K. Gomez, and G. Russello, “SMART: Shared memory based SDN architecture to resist DDoS ATtacks,” *ICETE 2020*

- *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications*, vol. 3, no. Icete, pp. 608–617, 2020, doi: 10.5220/0009864906080617.

[17] M. Fischer, “SDN / NFV-based DDoS Mitigation via Pushback,” 2020.

[18] M. Paliwal and K. K. Nagwanshi, “Effective Flow Table Space Management Using Policy-Based Routing Approach in Hybrid SDN Network,” *IEEE Access*, vol. 10, pp. 59806–59820, 2022, doi: 10.1109/ACCESS.2022.3180333.

[19] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015, doi: 10.1109/JPROC.2014.2371999.

[20] R. H. Jhaveri, R. Tan, and S. v. Ramani, “Real-time QoS Routing Scheme in SDN-based Robotic Cyber-Physical Systems QoS Routing with SDN for Manufacturing Robotics,” *ArXiv*, 2020.

[21] D. M. G. S. N. Ahuja, “dataset_sdn.” doi: 10.17632/jxpfjc64kr.1.

[22] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.

[23] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, no. 2, pp. 197–227, 2016, doi: 10.1007/s11749-016-0481-7.

[24] S. M. Lundberg, P. G. Allen, and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Adv Neural Inf Process Syst*, vol. 30, 2017, Accessed: Dec. 02, 2022. [Online]. Available: https://github.com/slundberg/shap

[25] A. N. Mazumder, N. Lyons, A. Dubey, A. Pandey, and A. Santra, “XAI-Increment: A Novel Approach Leveraging LIME Explanations for Improved Incremental Learning,” Nov. 2022, doi: 10.48550/arxiv.2211.01413.

[26] P. Biecek, “Dalex: Explainers for complex predictive models in R,” *Journal of Machine Learning Research*, vol. 19, pp. 1–5, 2018.

[27] W. Yang, H. Le, T. Laud, S. Savarese, and S. C. H. Hoi, “OmniXAI: A Library for Explainable AI,” Jun. 2022, Accessed: Dec. 02, 2022. [Online]. Available: http://arxiv.org/abs/2206.01612

[28] H. Scheers and T. de Laet, “Interactive and Explainable Advising Dashboard Opens the Black Box of Student Success Prediction,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12884 LNCS, pp. 52–66, 2021, doi: 10.1007/978-3-030-86436-1_5/COVER.

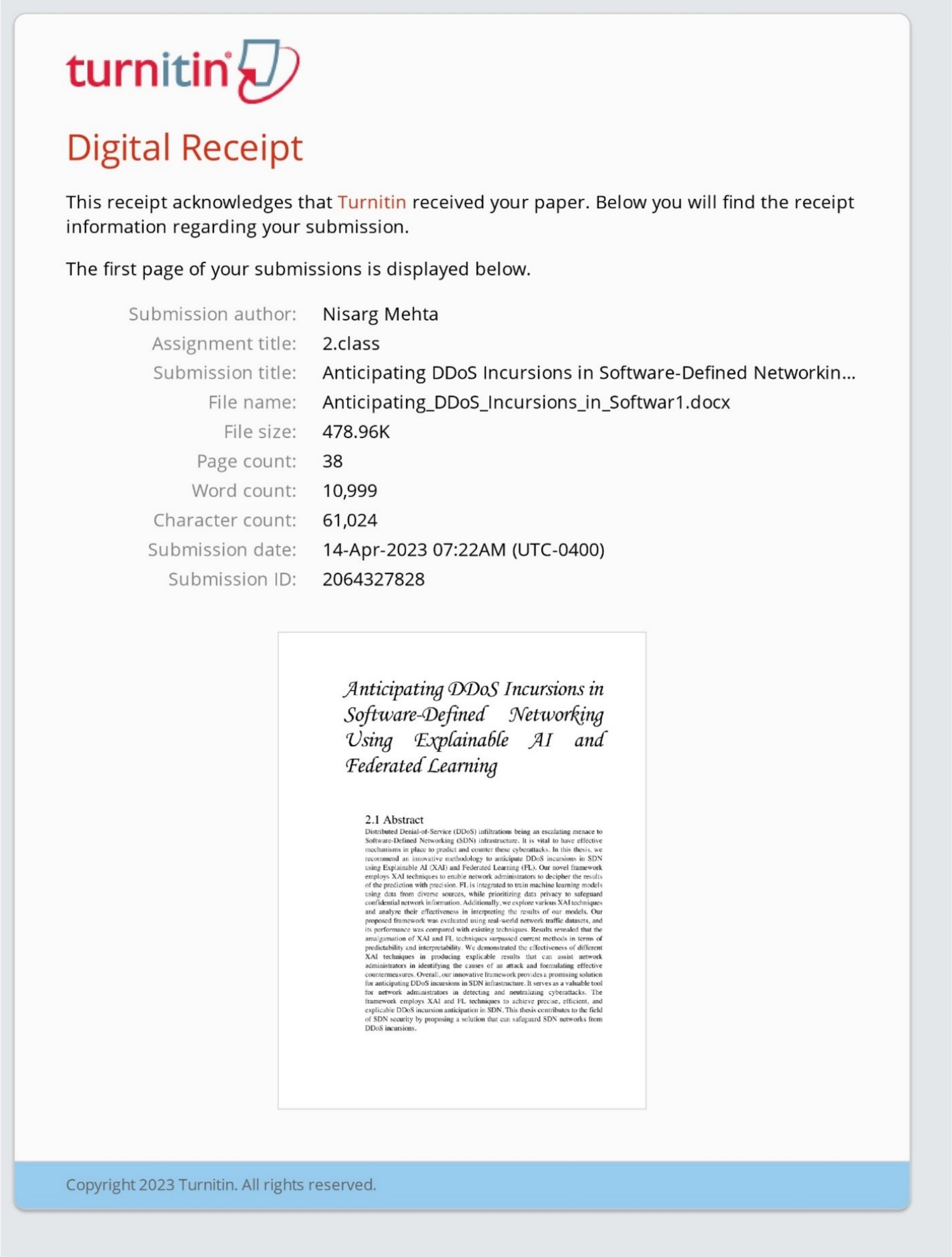
[29] W. Yang, H. Le, T. Laud, S. Savarese, and S. C. H. Hoi, “OmniXAI: A Library for Explainable AI,” Jun. 2022, doi: 10.48550/arxiv.2206.01612.

[30] “[2206.01612] OmniXAI: A Library for Explainable AI.” https://arxiv.org/abs/2206.01612 (accessed Dec. 02, 2022).

[31] “Explainable Boosting Machine.” https://interpret.ml/docs/ebm.html (accessed Dec. 02, 2022).

[32] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, “ELI5: Long Form Question Answering,” *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 3558–3567, Jul. 2019, doi: 10.48550/arxiv.1907.09190.

plagiarism receipt



turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

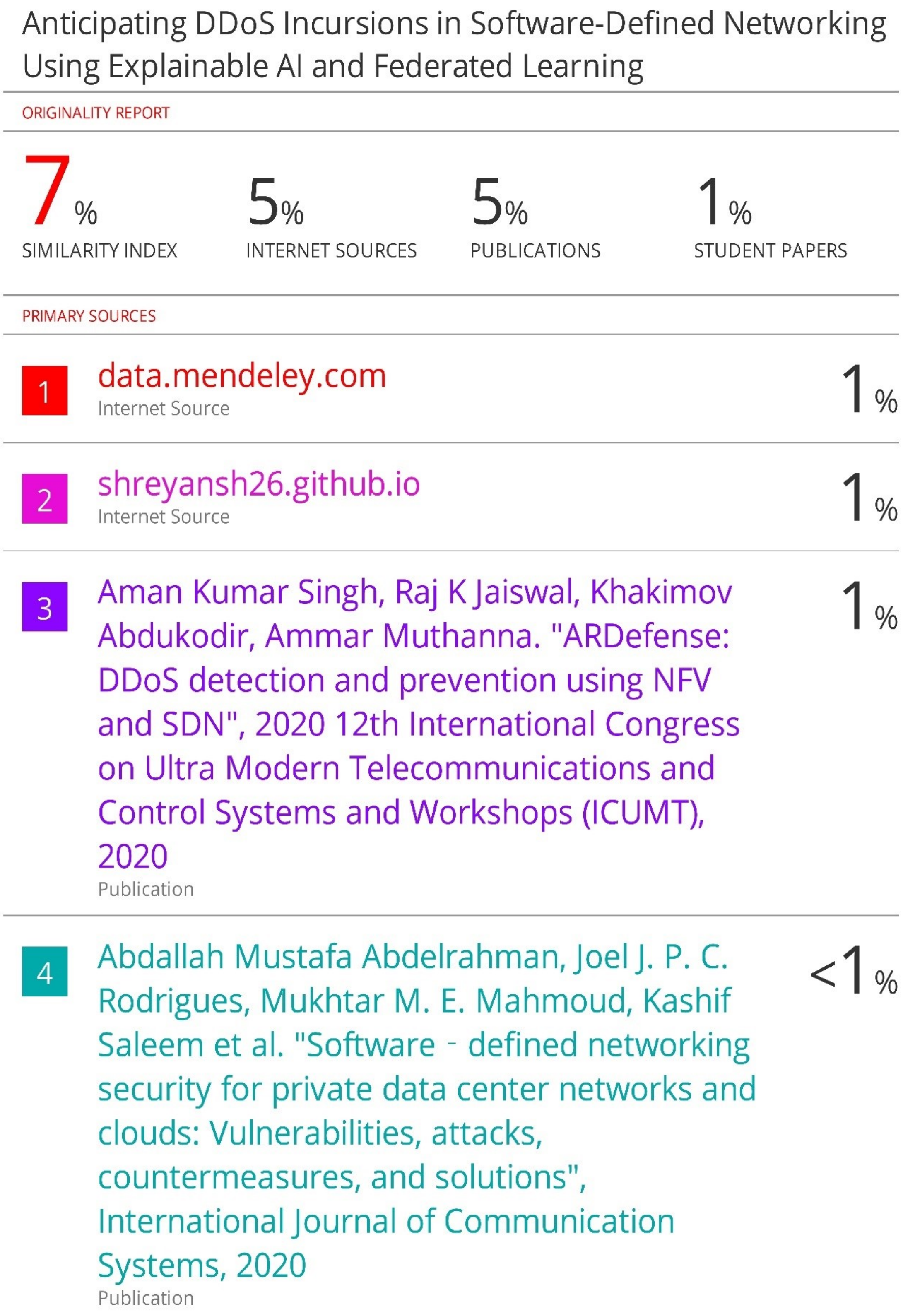
Submission author:	Nisarg Mehta
Assignment title:	2.class
Submission title:	Anticipating DDoS Incursions in Software-Defined Networkin...
File name:	Anticipating_DDoS_Incursions_in_Softwar1.docx
File size:	478.96K
Page count:	38
Word count:	10,999
Character count:	61,024
Submission date:	14-Apr-2023 07:22AM (UTC-0400)
Submission ID:	2064327828

Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning

2.1 Abstract

Distributed Denial-of-Service (DDoS) infiltrations being an escalating menace to Software-Defined Networking (SDN) infrastructure. It is vital to have effective mechanisms in place to predict and counter these cyberattacks. In this thesis, we recommend an innovative methodology to anticipate DDoS incursions in SDN using Explainable AI (XAI) and Federated Learning (FL). Our novel framework employs XAI techniques to enable network administrators to decipher the results of the prediction with precision. FL is integrated to train machine learning models using data from diverse sources, while prioritizing data privacy to safeguard confidential network information. Additionally, we explore various XAI techniques and analyze their effectiveness in interpreting the results of our models. Our proposed framework was evaluated using real-world network traffic datasets, and its performance was compared with existing techniques. Results revealed that the amalgamation of XAI and FL techniques surpassed current methods in terms of predictability and interpretability. We demonstrated the effectiveness of different XAI techniques in producing explicable results that can assist network administrators in identifying the causes of an attack and formulating effective countermeasures. Overall, our innovative framework provides a promising solution for anticipating DDoS incursions in SDN infrastructure. It serves as a valuable tool for network administrators in detecting and neutralizing cyberattacks. The framework employs XAI and FL techniques to achieve precise, efficient, and explicable DDoS incursion anticipation in SDN. This thesis contributes to the field of SDN security by proposing a solution that can safeguard SDN networks from DDoS incursions.

similarity index which include content from abstract to references.



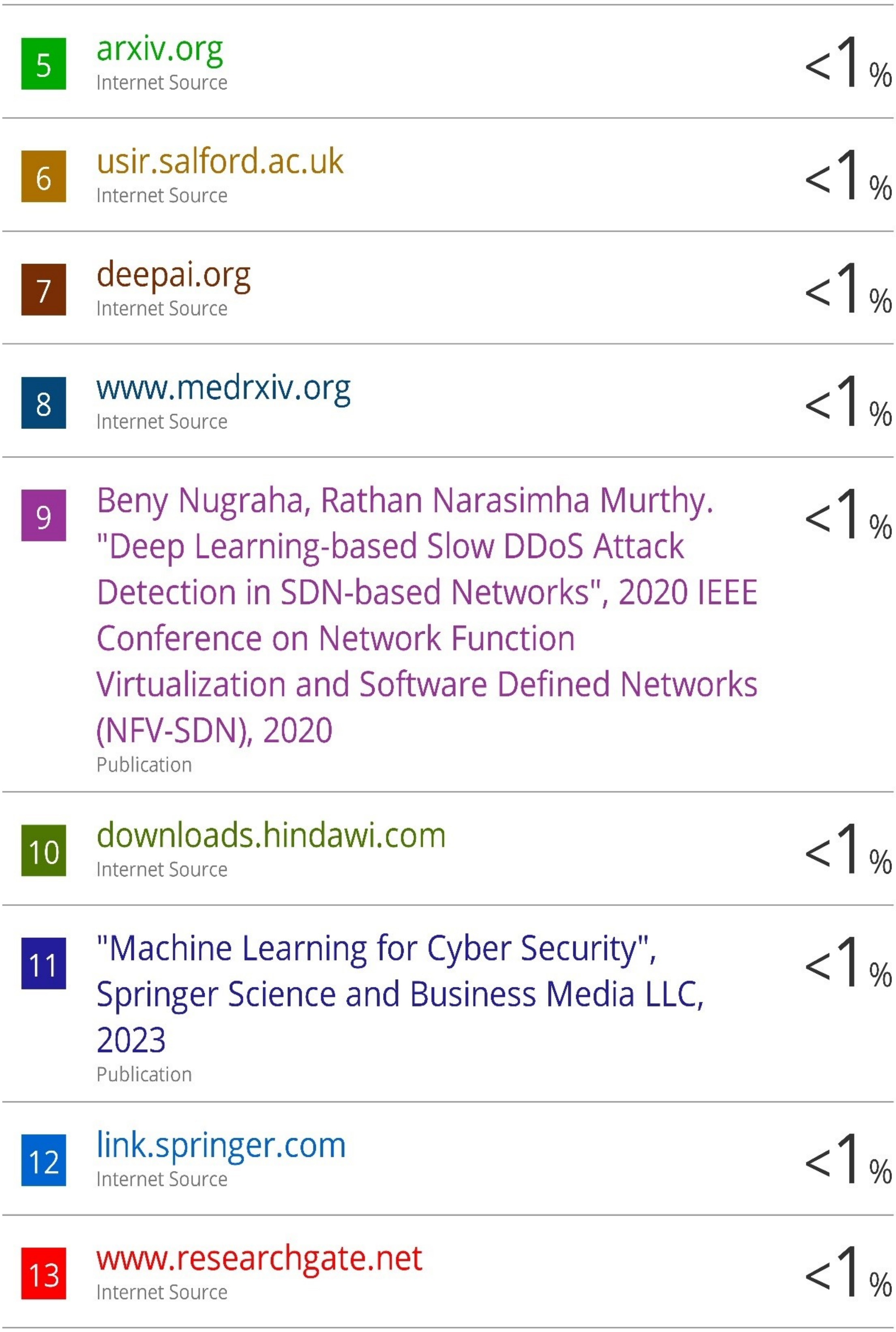
Anticipating DDoS Incursions in Software-Defined Networking Using Explainable AI and Federated Learning

ORIGINALITY REPORT

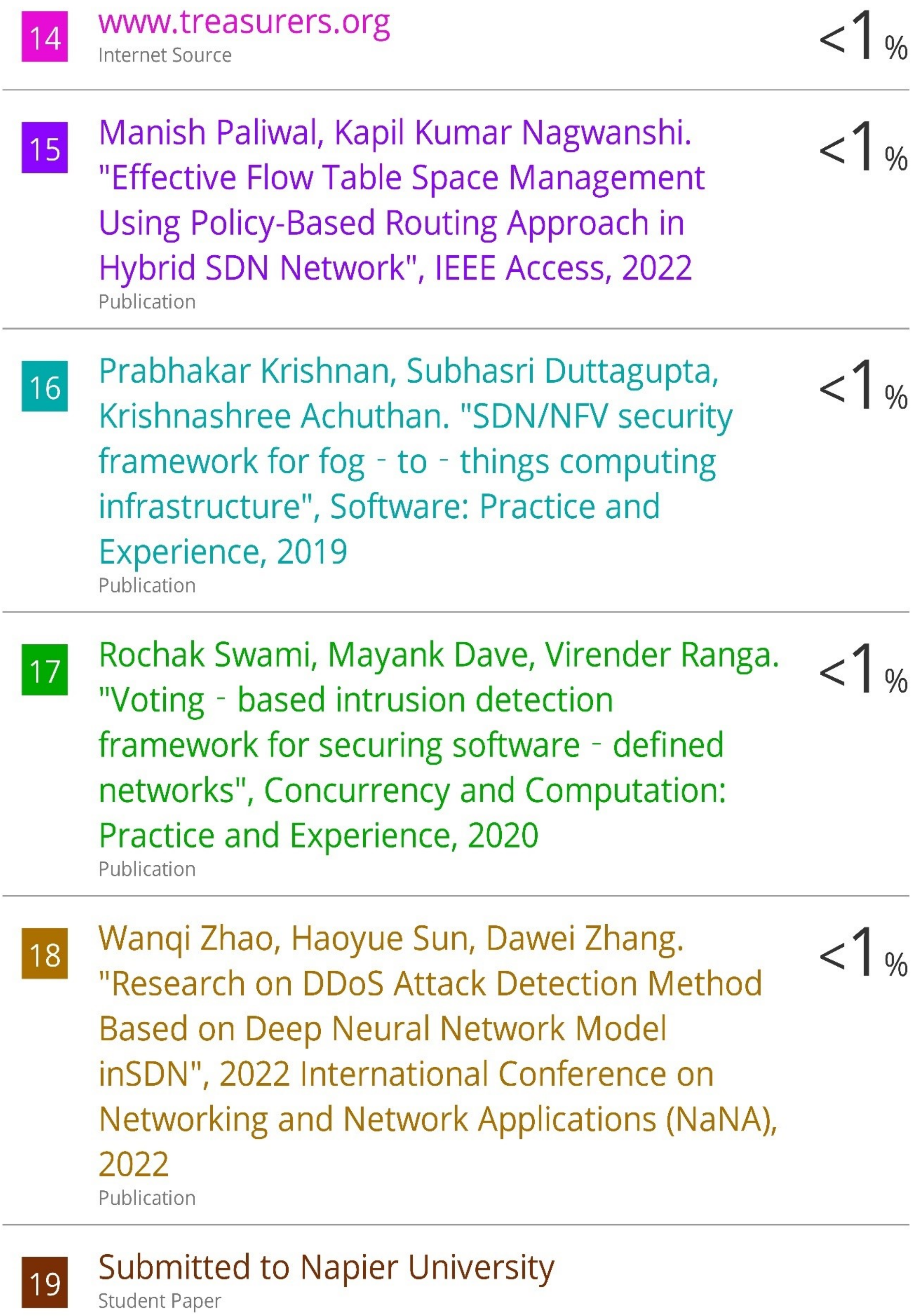
7%	5%	5%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

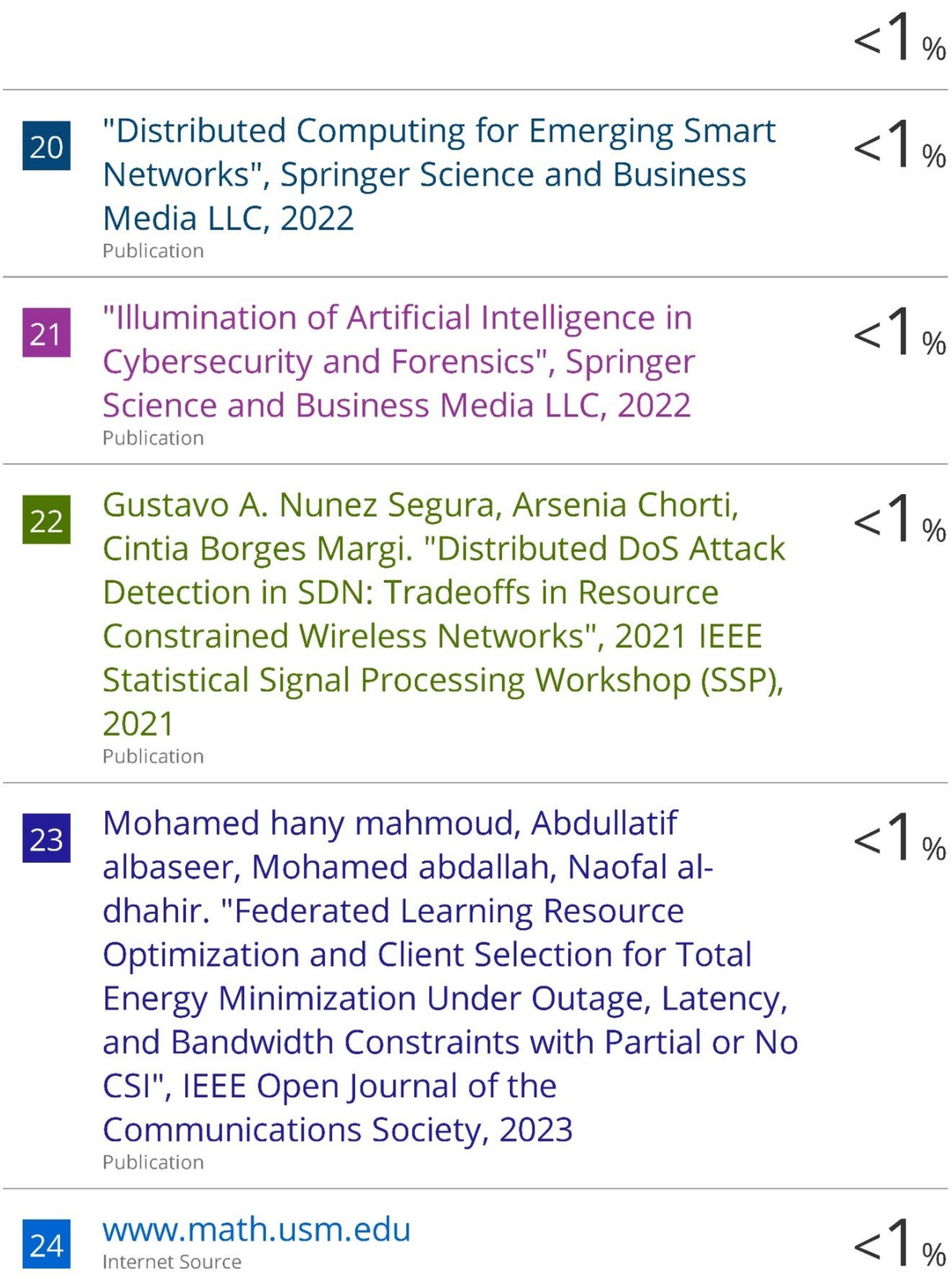
1	data.mendeley.com Internet Source	1%
2	shreyansh26.github.io Internet Source	1%
3	Aman Kumar Singh, Raj K Jaiswal, Khakimov Abdukodir, Ammar Muthanna. "ARDefense: DDoS detection and prevention using NFV and SDN", 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2020 Publication	1%
4	Abdallah Mustafa Abdelrahman, Joel J. P. C. Rodrigues, Mukhtar M. E. Mahmoud, Kashif Saleem et al. "Software‐defined networking security for private data center networks and clouds: Vulnerabilities, attacks, countermeasures, and solutions", International Journal of Communication Systems, 2020 Publication	<1%



5	arxiv.org Internet Source	<1 %
6	usir.salford.ac.uk Internet Source	<1 %
7	deepai.org Internet Source	<1 %
8	www.medrxiv.org Internet Source	<1 %
9	Beny Nugraha, Rathan Narasimha Murthy. "Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks", 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2020 Publication	<1 %
10	downloads.hindawi.com Internet Source	<1 %
11	"Machine Learning for Cyber Security", Springer Science and Business Media LLC, 2023 Publication	<1 %
12	link.springer.com Internet Source	<1 %
13	www.researchgate.net Internet Source	<1 %



14	www.treasurers.org Internet Source	<1%
15	Manish Paliwal, Kapil Kumar Nagwanshi. "Effective Flow Table Space Management Using Policy-Based Routing Approach in Hybrid SDN Network", IEEE Access, 2022 Publication	<1%
16	Prabhakar Krishnan, Subhasri Duttagupta, Krishnashree Achuthan. "SDN/NFV security framework for fog - to - things computing infrastructure", Software: Practice and Experience, 2019 Publication	<1%
17	Rochak Swami, Mayank Dave, Virender Ranga. "Voting - based intrusion detection framework for securing software - defined networks", Concurrency and Computation: Practice and Experience, 2020 Publication	<1%
18	Wanqi Zhao, Haoyue Sun, Dawei Zhang. "Research on DDoS Attack Detection Method Based on Deep Neural Network Model inSDN", 2022 International Conference on Networking and Network Applications (NaNA), 2022 Publication	<1%
19	Submitted to Napier University Student Paper	



<1%

20 "Distributed Computing for Emerging Smart Networks", Springer Science and Business Media LLC, 2022
Publication
<1%

21 "Illumination of Artificial Intelligence in Cybersecurity and Forensics", Springer Science and Business Media LLC, 2022
Publication
<1%

22 Gustavo A. Nunez Segura, Arsenia Chorti, Cintia Borges Margi. "Distributed DoS Attack Detection in SDN: Tradeoffs in Resource Constrained Wireless Networks", 2021 IEEE Statistical Signal Processing Workshop (SSP), 2021
Publication
<1%

23 Mohamed hany mahmoud, Abdullatif albaseer, Mohamed abdallah, Naofal al-dhahir. "Federated Learning Resource Optimization and Client Selection for Total Energy Minimization Under Outage, Latency, and Bandwidth Constraints with Partial or No CSI", IEEE Open Journal of the Communications Society, 2023
Publication
<1%

24 www.math.usm.edu
Internet Source
<1%

25	Aastha Maheshwari, Burhan Mehraj, Mohd Shaad Khan, Mohd Shaheem Idrisi. "An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment", Microprocessors and Microsystems, 2022 Publication	<1 %
26	Franciscus X.A. Wibowo, Mark A. Gregory, Khandakar Ahmed, Karina M. Gomez. "Multi-domain Software Defined Networking: Research status and challenges", Journal of Network and Computer Applications, 2017 Publication	<1 %
27	www.biorxiv.org Internet Source	<1 %
28	www.infocommunications.hu Internet Source	<1 %
29	www.mdpi.com Internet Source	<1 %
30	"Intelligent Interactive Multimedia Systems for e-Healthcare Applications", Springer Science and Business Media LLC, 2022 Publication	<1 %
31	Jahanzaib Malik, Adnan Akhunzada, Iram Bibi, Muhammad Talha, Mian Ahmad Jan, Muhammad Usman. "Security-aware Data-	<1 %

	driven Intelligent Transportation Systems", IEEE Sensors Journal, 2020 Publication	
32	Nagarathna Ravi, S Mercy Shalinie. "Learning Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud architecture", IEEE Internet of Things Journal, 2020 Publication	<1 %
33	Song Wang, Juan Fernando Balarezo, Karina Gomez Chavez, Akram Al-Hourani et al. "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques", Engineering Science and Technology, an International Journal, 2022 Publication	<1 %
34	insightsociety.org Internet Source	<1 %
35	pingpdf.com Internet Source	<1 %
36	ve.co-aol.com Internet Source	<1 %
37	www.misa.net.au Internet Source	<1 %
38	Saptarshi Debroy, Prasad Calyam, Minh Nguyen, Roshan Lal Neupane, Bidyut Mukherjee, Ajay Kumar Eeralla, Khaled Salah. "Frequency-Minimal Utility-Maximal Moving	<1 %

Target Defense against DDoS in SDN-based Systems", IEEE Transactions on Network and Service Management, 2020

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		