ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE

№1
2026

ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)

CENTRAL ASIAN ACADEMIC
RESEARCH CENTER

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE

1 (357)

JANUARY – MARCH 2026

PUBLISHED SINCE JANUARY 1963
PUBLISHED 4 TIMES A YEAR

ALMATY, NAS RK

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Owner: «Central Asian Academic Research Center» LLP (Almaty).
Certificate **№ KZ77VPY00121154** on the re-registration of the periodical printed and online publication of the information agency, issued on **05.06.2025** by the Republican State Institution «Information Committee» of the Ministry of Culture and Information of the Republic of Kazakhstan
Subject area*: information and communication technologies.*
Currently: *included in the list of journals recommended by the CCSES MSHE RK in the direction of «Information and communication technologies».*
Periodicity: *4 times a year.*
http://www.physico-mathematical.kz/index.php/en/

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Меншіктеуші: «Орталық Азия академиялық ғылыми орталығы» ЖШС (Алматы).
Ақпарат агенттігінің мерзімді баспасөз басылымын, ақпарат агенттігін және желілік басылымды қайта есепке қою туралы ҚР Мәдениет және Ақпарат министрлігі «Ақпарат комитеті» Республикалық мемлекеттік мекемесі **05.06.2025** ж. берген **№ KZ77VPY00121154** Куәлік.
Тақырыптық бағыты: *ақпараттық-коммуникациялық технологиялар*
Қазіргі уақытта: *«ақпараттық-коммуникациялық технологиялар» бағыты бойынша ҚР БҒМ БҒСБК ұсынған журналдар тізіміне енді.*
Мерзімділігі: *жылына 4 рет.*
http://www.physico-mathematical.kz/index.php/en/

Academic Scientific Journal of Computer Science
ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)
Собственник: *ТОО «Центрально-азиатский академический научный центр» (г. Алматы).*
Свидетельство о постановке на переучет периодического печатного издания, информационного агентства и сетевого издания **№ KZ77VPY00121154.** Дата выдачи **05.06.2025**
Тематическая направленность: *информационно-коммуникационные технологии.*
В настоящее время: *вошел в список журналов, рекомендованных КОКСНВО МНВО РК по направлению «информационно-коммуникационные технологии».*
Периодичность: *4 раза в год.*
http://www.physico-mathematical.kz/index.php/en/

CONTENTS

COMPUTER SCIENCE

МАЗМҰНЫ

ИНФОРМАТИКА

СОДЕРЖАНИЕ

ИНФОРМАТИКА

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE
ISSN 1991-346X
Volume 1.
Namber 357 (2026). 30–51

https://doi.org/10.32014/2026.2518-1726.400

IRSTI 20.15.05
UDC 004.7

© Amanov A.N.*, Kazbekova G.N., Zhunissov N.M., Abibullayeva A.A., Aben A.B., 2026.
Khoja Akhmet Yassawi International Kazakh-Turkish University,
Turkistan, Kazakhstan.
E-mail: anuarbek.amanov@ayu.edu.kz

ARTIFICIAL INTELLIGENCE-BASED INTRUSION DETECTION FOR DDOS ATTACKS IN SOFTWARE DEFINED NETWORKING

Amanov Anuarbek — PhD, Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkestan, Kazakhstan,
E-mail: anuarbek.amanov@ayu.edu.kz, https://orcid.org/0000-0003-0638-6859;
Kazbekova Gulnur — Candidate of Technical Sciences, Associate Professor, International Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkestan, Kazakhstan,
E-mail: gulnur.kazbekova@ayu.edu.kz, https://orcid.org/0000-0002-2756-7926;
Zhunissov Nurseit — PhD, Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkestan, Kazakhstan,
E-mail: nurseit.zhunissov@ayu.edu.kz, https://orcid.org/0000-0001-6531-9408;
Abibullayeva Aiman — PhD, Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkestan, Kazakhstan,
E-mail: aiman.abibullayeva@ayu.edu.kz, https://orcid.org/(0000-0003-2449-2540;
Aben Arypzhan — Doctoral student in the educational program «Information systems», Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkestan, Kazakhstan,
E-mail: arypzhan.aben@ayu.edu.kz, https://orcid.org/0000-0001-8534-3288.

Abstract. Software Defined Networks (SDN) are a network architecture that allows for the rapid deployment of services by centrally programming the control and data planes (Control Plane and Data Plane) by forwarding/switching specific packets. However, the presence of a central controller node and open northbound/southbound interfaces can lead to control channel overload, overflow of flow tables, and service outages during distributed denial-of-service (DDoS) attacks. To detect and mitigate DDoS attacks in an SDN environment in real time, a flexible modular software architecture was developed and a machine learning-based detector was integrated into it. First, normal traffic was monitored for a long time, and features such as flow count, packet/byte rate, port load, and frequency of new flows were extracted from OpenFlow statistics. Then, many scenarios were run on the Mininet emulator for different topologies, attack intensity, and background load, and a set of

defined data sets were created. In the Google Collaboratory environment, Decision Tree, Artificial Neural Network (ANN), and Naive Bayes models were trained and compared through cross-validation; the best model was re-integrated into the detection and response modules at the controller level. Decision Tree showed 91% accuracy, outperforming ANN (78%) and Naive Bayes (68%). The system detected the anomaly at an early stage and mitigated the impact of the attack by automatically implementing blocking or rate limiting rules for malicious IP/port/flows through the controller. The modular approach allows for rapid deployment of new AI models, saving training time, and adapting to real-world SDN infrastructures.

Keywords: DDoS, SDN, Artificial Intelligence, Machine Learning, Network Security, Decision Tree model

For citations: Amanov A.N., Kazbekova G.N., Zhunissov N.M., Abibullayeva A.A., Aben A.B. Artificial intelligence-based intrusion detection for DDOS attacks in software defined networking. Academic Scientific Journal of Computer Science, 2026. — No.1. – P. 30–51 . DOI: https://doi.org/10.32014/2026.2518-1726.400

© Аманов А.Н. *, Казбекова Г.Н., Жунисов Н.М., Абибуллаева А.А., Абен А.Б., 2026.
Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті,
Түркістан, Қазақстан.
E-mail: anuarbek.amanov@ayu.edu.kz

БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМАМЕН АНЫҚТАЛҒАН ЖЕЛІДЕГІ DDOS ШАБУЫЛДАРЫ ҮШІН ЖАСАНДЫ ИНТЕЛЛЕКТКЕ НЕГІЗДЕЛГЕН ШАБУЫЛДАРДЫ АНЫҚТАУ

Аманов Ануарбек — PhD, аға оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан,
E-mail: anuarbek.amanov@ayu.edu.kz, https://orcid.org/0000-0003-0638-6859;
Казбекова Гулнур — техника ғылымдарының кандидаты, доцент м.а., Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан,
E-mail: gulnur.kazbekova@ayu.edu.kz, https://orcid.org/0000-0002-2756-7926;
Жунисов Нурсейт — PhD, аға оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан,
E-mail: nurseit.zhunissov@ayu.edu.kz, https://orcid.org/0000-0001-6531-9408;
Абибуллаева Айман — PhD, аға оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан,
E-mail: aiman.abibullayeva@ayu.edu.kz, https://orcid.org/(0000-0003-2449-2540;
Абен Арыпжан —«Ақпараттық жүйелер» білім беру бағдарламасы бойынша докторант, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан,
E-mail: arypzhan.aben@ayu.edu.kz, https://orcid.org/0000-0001-8534-3288.

Аннотация. Бағдарламалық жасақтамамен анықталған желілер (SDN-Software Defined Network) басқару және деректер жазықтықтарын бөлу

(Басқару жазықтығы (Control Plane) және Деректер жазықтығы (Data Plane)) нақты пакеттерді өткізеді (forwarding/switching)) арқылы желіні орталықтандырылған түрде бағдарламалап, сервистерді жылдам енгізуге жағдай жасайды. Дегенмен контроллердің орталық түйін болуы және ашық northbound/southbound интерфейстері таратылған қызмет көрсетуден бас тарту (DDoS) шабуылдары кезінде басқару арнасының шамадан тыс жүктелуіне, ағын кестелерінің толып кетуіне және қызметтің тоқтауына әкелуі мүмкін. SDN ортасында DDoS шабуылдарын нақты уақытта анықтау және әлсірету үшін икемді модульдік бағдарламалық архитектураны әзірлеу және оған машинамен оқытуға негізделген детекторды біріктіру. Алдымен қалыпты трафик ұзақ уақыт бақыланып, OpenFlow статистикасынан ағын саны, пакет/байт жылдамдығы, порттық жүктеме, жаңа ағындардың пайда болу жиілігі сияқты белгілер алынды. Кейін Mininet эмуляторында әртүрлі топологиялар, шабуыл қарқындылығы және фондық жүктеме бойынша көптеген сценарийлер орындалып, белгіленген деректер жиыны құрылды. Google Colaboratory ортасында Decision Tree, жасанды нейрондық желі (ANN) және Naive Bayes модельдері оқытылып, кросс-валидация арқылы салыстырылды; үздік модель контроллер деңгейінде детекция және жауап модульдеріне қайта біріктірілді. Decision Tree 91% дәлдік көрсетіп, ANN (78%) және Naive Bayes (68%) көрсеткіштерінен жоғары болды. Жүйе аномалияны ерте кезеңде анықтап, контроллер арқылы зиянды IP/порт/ағындарға қатысты блоктау немесе жылдамдықты шектеу ережелерін автоматты енгізу арқылы шабуыл әсерін азайтты. Модульдік тәсіл жаңа ЖИ модельдерін тез ауыстыруға, оқыту уақытын үнемдеуге және нақты өндірістік SDN инфрақұрылымына бейімдеуге мүмкіндік береді.

Түйін сөздер: DDoS, SDN, жасанды интеллект, машиналық оқыту, желілік қауіпсіздік, Decision Tree

© Аманов А.Н. *, Казбекова Г.Н., Жунисов Н.М., Абибуллаева А.А., Абен А.Б., 2026.

Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан.

E-mail: anuarbek.amanov@ayu.edu.kz

ОБНАРУЖЕНИЕ ВТОРЖЕНИЙ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ DDOS-АТАК В ПРОГРАММНО-ОПРЕДЕЛЯЕМЫХ СЕТЯХ

Аманов Ануарбек — PhD, старший преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан,
E-mail: anuarbek.amanov@ayu.edu.kz, https://orcid.org/0000-0003-0638-6859;
Казбекова Гулнур — кандидат технических наук, доцент, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан,
E-mail: gulnur.kazbekova@ayu.edu.kz, https://orcid.org/0000-0002-2756-7926;

Жунисов Нурсейт — PhD, старший преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан,
E-mail: nurseit.zhunissov@ayu.edu.kz, https://orcid.org/0000-0001-6531-9408;
Абибуллаева Айман — PhD, старший преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан,
E-mail: aiman.abibullayeva@ayu.edu.kz, https://orcid.org/0000-0003-2449-2540;
Абен Арыпжан Бактиярович – докторант по ОП «Информационные системы», Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, Туркестан, Казахстан,
E-mail: arypzhan.aben@ayu.edu.kz, https://orcid.org/0000-0001-8534-3288.

Аннотация. Программно-определяемые сети (SDN) представляют собой современную сетевую архитектуру, позволяющую оперативно развертывать сервисы за счёт централизованного управления плоскостями управления и данных посредством программного контроля обработки пакетов. Однако наличие центрального узла управления и открытых интерфейсов северного и южного направлений может приводить к перегрузке каналов управления, переполнению таблиц потоков и сбоям в работе сервисов в условиях распределённых атак типа «отказ в обслуживании» (DDoS). Для обнаружения и смягчения DDoS-атак в среде SDN в режиме реального времени разработана гибкая модульная программная архитектура с интеграцией детектора на основе методов машинного обучения. На первом этапе осуществлялся мониторинг нормального сетевого трафика, на основе которого из статистики OpenFlow извлекались ключевые характеристики, включая количество потоков, скорость передачи пакетов и байтов, загрузку портов и частоту появления новых потоков. Далее в эмуляционной среде Mininet были реализованы различные сценарии с вариацией топологий сети, интенсивности атак и фоновой нагрузки, что позволило сформировать репрезентативный набор данных для обучения моделей. В среде Google Colab были обучены и сравнены модели дерева решений, искусственной нейронной сети и наивного байесовского классификатора с использованием перекрёстной проверки. По результатам экспериментов наилучшие показатели продемонстрировала модель дерева решений с точностью 91%, превзойдя искусственную нейронную сеть (78%) и наивный байесовский классификатор (68%). Разработанная система обеспечивает раннее обнаружение аномалий и автоматическое реагирование за счёт внедрения правил блокировки или ограничения скорости для вредоносных IP-адресов, портов и потоков на уровне контроллера. Модульный подход обеспечивает возможность оперативного внедрения новых моделей искусственного интеллекта, сокращает время обучения и способствует адаптации системы к реальным условиям функционирования SDN-инфраструктур.

Ключевые слова: DDoS, SDN, Искусственный интеллект, Машинное обучение, Сетевая безопасность, Модель Decision Tree

Introduction. A Computer Network is defined as the system formed by connecting devices with the purpose of sharing information and resources within certain rules. As mentioned earlier, the main purpose of a computer network is to share information and resources. Computer networks are classified based on the area they cover, the communication technologies they employ, and their network structures.

Innovations and improvements in the field of information and communication technology, which have become an inseparable part of our everyday lives, have accelerated over the past 30 years. With the widespread use of the internet, the exchange of information has quickened, and technological developments have advanced significantly. The existing Traditional Network Architecture is insufficient to meet the demands of rapidly evolving global technologies. However, managing Traditional Network Architecture is a demanding and complex task. This architecture primarily consists of the infrastructure that links network systems and the devices connected to this infrastructure. Passive components enable computers to communicate with one another within these systems, while electronic devices facilitate communication across the infrastructure. Examples of such electronic equipment include routers, switches, and other intermediary internet devices. Traditional Network Architecture also encompasses several complex protocols. While each new protocol may solve one issue, it often introduces another. Moreover, non-standard network devices, communication systems, and network management software from various manufacturers can operate the network and support diverse applications running on it. The introduction of a non-standard network device into the system can lead to major security flaws. For these reasons, Traditional Network Architecture is complex and difficult to manage. To address the aforementioned network issues, the concept of SDN has evolved. SDN emerges as a new network approach that proposes the separation of data and control planes, which are integrated into the infrastructure of Traditional Network Architecture. This separation simplifies the network and its management systems, providing an autonomous structure in an attempt to eliminate the problems associated with today's network infrastructure. OpenFlow is an open standards-based protocol that enables communication between SDN controllers and network devices. The controller and OpenFlow-enabled network switches are the most significant components of OpenFlow. They facilitate a centralized control plane that enables the development of a dynamic, scalable, efficient, and high-performance communication infrastructure. In addition to these advantages, SDN introduces new security concerns.

Controller coding must be done correctly for SDN to function properly. The number of controllers, for example, is a critical factor in network optimization scaling. For large-scale networks, a single SDN controller also poses issues with scalability, response time, infrastructure support, and availability. A single controller's limited capacity will not be sufficient to handle the huge data flows of tens of thousands of nodes in large networks. Finally, consideration should be given

to SDN security. When SDN is insecure, the network can be subjected to a variety of assaults.

Network attacks are classified into two types: active and passive. Active attacks are motivated by the attackers' desire to obtain the information they want. If there is a system that monitors their behavior, anything they do to visit the site will be noticed. However, if attackers are unaware of what to look for, they can carry out the attack unnoticed. Passive attacks, on the other hand, seek to obtain further information. The majority of attempts take the form of both active and passive attacks. Active attacks are the easiest to identify, yet the majority go undiscovered. Passive attacks are more difficult to detect. As a result, the chances of detecting the passive attack are almost nonexistent.

In many circumstances, attackers' resort to active attacks first in order to get access to the desired network site. Then, using a passive attack, they gain access to the information they seek. Phishing, password attacks, DoS and DDoS attacks are some examples. The most significant active attacks are denial of service attacks and hacking. Denial of service occurs when typical system users are denied service (Raktate et al., 2024). This includes everything from prohibiting people from accessing a private website to deactivating user accounts that allow them to log into the network. Any network connected to the Internet is subject to denial-of-service attacks. The tools required to carry out this type of assault are easily accessible, and such programs are not difficult to run.

DDoS attacks are a type of distributed network attack. These attacks take advantage of capacity limitations that apply to any network resource, such as the infrastructure that hosts a company's website. By making numerous requests to the attacked web resource, a DDoS assault seeks to overwhelm the website's capacity to handle huge amounts of requests and prevent it from working effectively. The ease with which DDoS assaults can be carried out at cheap cost poses a significant risk to firms who conduct their business on the internet. Organizations who are caught off guard by this form of DDoS attack may be unable to serve for an extended period of time.

Attackers often employ three types of DDoS assault tactics (network, protocol, and application). DDoS attacks on networks are the most prevalent sort of assault utilized by attackers, with the goal of consuming the internet bandwidth used by target companies. As a result, both network traffic entering and exiting the organization will be affected, and both will be unable to respond to genuine requests and will be unavailable. As a result, all parties who use the internet infrastructure will be impacted, and all internet-dependent services would be rendered inoperable. TCP/UDP flood, DNS/NTP/Memcached Amplification assaults are examples of Network Type DDoS Attacks.

Protocol assaults are broad attacks against OSI K3/K4 protocols. Targeted systems include firewalls, load balancing devices, routers, and other devices that use connection session information. Multiple login requests are submitted, and additional requests are sent before the session ends. As a result, network and

security devices will fill the session tables and become dysfunctional. Protocol assaults include SYN/SYN-ACK/ACK flood, ping of death, and others.

Application attacks include web apps, DNS, SMTP, and so on. OSI K7 attacks target application services. The main purpose is to destabilize the system by delivering requests to applications that are much over their capability and utilizing their resources. Overflow attacks on HTTP, HTTPS, DNS, and SMTP services are examples of application attacks.

Unfortunately, there is currently no conclusive and permanent strategy to avoid becoming a target of DDoS assaults. However, there are several measures that can lessen the likelihood of becoming a target as well as the severity of attacks. Separating DDoS attacks from immediate and typical system performance boosts and drops necessitates the use of the appropriate technologies and experience. In terms of businesses, among the major safeguards are a well-designed network architecture and a high degree of system and TCP/IP understanding among necessary staff. If router-level protection is given, packets destined to target systems are routed through the router before being forwarded to other systems. With this feature, routers are the first systems to be attacked, and the measures taken over the routers are critical in terms of fighting the attack from the start. If some router settings and the features of the incoming packets during the attack can be determined, the attacks can be stopped or their impacts limited with the access control list that will be established.

Considering the risks of a network attack, software or hardware tools such as firewalls, antivirus, or Intrusion Detection Systems (IDS) have been built to thwart internet-based attacks. Cyber security professionals utilize security software to secure their data and systems from malicious network users. Relying just on a firewall mechanism to protect network systems is insufficient. As a result, an IDS should be utilized in addition to the firewall. IDS are systems that automatically examine events in a computer network to detect external threats. The only way for IDS to keep up with the speed of emerging attack strategies is for the system to be constantly updated. Companies are concerned about systems that have been disconnected from the internet for an extended period of time and have not been updated. This adds to the workload of network security professionals.

This article describes a customizable modular architecture for detecting and preventing DDoS attacks. In SDN, normal traffic between computers was monitored. Then a DDoS attack was started, which was followed by another attack. In this paper, a system is created to analyze and apply Machine Learning techniques to fight DDoS attacks in SDN. The system has been tested via Mininet utilizing various test scenarios. In the 2nd part of the article, the method used in the study is presented, in the 3rd part the findings are evaluated comprehensively and in the 4th part the results obtained are discussed.

Methodology. Selected Machine Learning Algorithms. Machine learning, a subset of artificial intelligence, detects complex patterns in data and makes informed conclusions using statistical methods and computational power. In

classification problems, machine learning approaches have been successfully applied (Gangadhar and Sterbenz, 2017; Sultana et al., 2019; Ray, 2024). presented a solution for detecting and analyzing DDoS (Distributed Denial of Service) assaults on cloud computing services using Dempster-Shafer Theory (DST) and Fault Tree Analysis (FTA) for a virtual machine (VM) intrusion detection system (IDS). This solution quantitatively represents uncertainty and is effectively used in intrusion detection systems to reduce false alarm rates. Common machine learning techniques employed in intrusion detection systems, as mentioned in the literature, include Bayesian classification, Support Vector Machine (SVM), decision tree, and artificial neural networks. A study (Peng et al., 2018) identified anomalous flows in SDN (Software Defined Networking) networks, highlighting the effectiveness of Machine Learning (ML) techniques in detecting malicious flows. These techniques enable switches to use information from flow tables to monitor information flow, processing unusual flows through a detection module equipped with the DPTCM-KNN algorithm. However, this process, repeated every ten seconds, introduces significant processing overhead as a potential issue. Additionally, another study proposed a trust-based method that uses the device profile and packet data to identify malicious devices, further enhancing security measures in SDN networks.

The used method based on Decision Tree, ANN, Naive Bayes machine learning algorithms and the main steps showed in Figure 1. The proposed matching strategy based on the matched the incoming request real-time from the nodes compared with the trained classifier stored data behavior as (SDN specific dataset generated by using mininet emulator and used for traffic classification by machine learning). In addition, the proposed trained model can classified as attack traffic if it is not matched packet details with the stored behavior, which matched, in the first step.

Bayesian networks are investigated in the context of guided learning in machine learning. There is usually a pattern at work in the Bayesian classification process, and this pattern determines the previously determined classes. As an example, consider the process of recognizing superfluous messages (spam) in incoming e-mails. Spam email and non-spam email are two classes in this example. An example of machine learning with instruction is an algorithm that will utilize the spam and non-spam e-mails we have and decide if the e-mails we will receive in the future are spam or not (Çalış et al., 2013). The Bayes theorem is based on determining which factor has a greater part in the occurrence of an event involving several factors. Equation 1 shows how Bayes' theorem can be represented.

Equations should be centered in the page. Equation number should be written inside paranthesis at the right side of the equation, and should be aligned right in the page. A sample equation, Equation 1 is shown below. Equations should be cited with their full names like "as seen in Equation 1", "as seen in Equations 3 and 4".

$$□(□|□) = □(□|□)\ □(□)\ □(□) \quad (1)$$

On Equation 1

P(e) = prior probability of event e,

P(T) = a priori probability of training data T

The conditional probability of T given the event P(T|e) = e

P(e|T) = T is expressed as the conditional probability of e given the training data.

In intrusion detection systems, Bayesian classifiers are a common strategy. The highest classification success with Bayesian analysis was achieved by (Panigrahi and Patra, 2019) using the DARPA '99 dataset in their IDS. The authors achieved 99.62 percent accuracy in DoS (Denial of Service) assaults, 100 percent accuracy in information scanning attacks, 98.63 percent accuracy in U2R (User to Root) attacks, and 42.62 percent accuracy in R2L (Remote to Local) attacks in their studies. (Alhakami et al., 2019) demonstrated a 99.82 percent in accuracy in identifying normal behaviors, 99.49 percent DoS assaults, 99.72 percent in information scanning attacks, 99.47 percent in U2R attacks, and 99.35 percent in R2L attacks using the KDD Cup '99 dataset. They were successful.

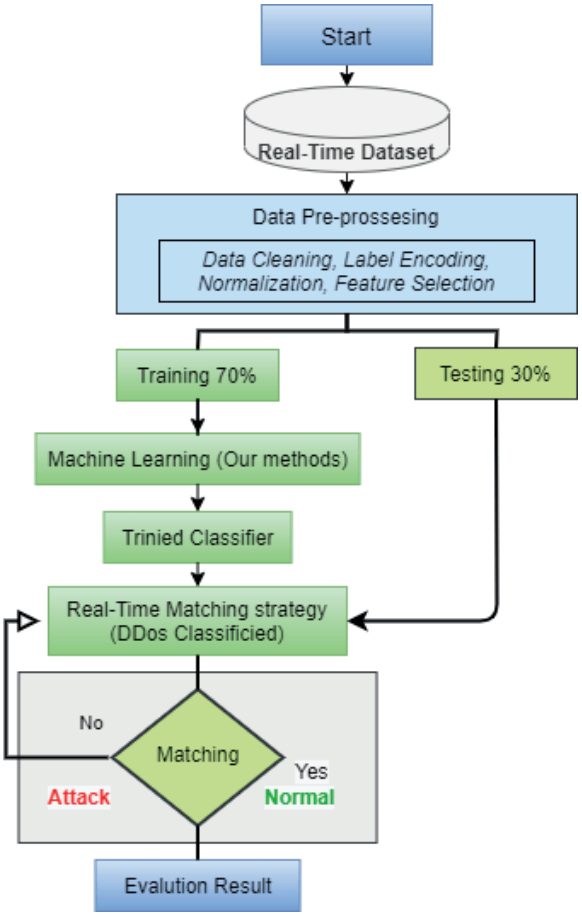

Figure 1 – Suggested machine learning system model

Huang et al., proposed the Support Vector Machine (SVM) as a powerful classifier in 2018. It is based on statistical methods. SVM is a supervised learning model that recognizes and analyzes patterns through classification and regression analysis (Huang et al., 2018). A labeled input dataset is required for SVM, which outputs two classes from the input dataset in binary classification problems. The training examples submitted fall into one of these two categories. The SVM training method creates a model to categorize new samples. The SVM model represents objects in space as points, separated by a hyperplane that is as wide as possible to differentiate the categories. New samples are positioned in this space, and their category is predicted based on which side of the hyperplane they fall on. SVM is widely used in intrusion detection systems.

Zhang, Y., and Zhu, Y. (2010) achieved high classification success with SVM in a 2010 study utilizing the KDD Cup '99 dataset, with 99.32 percent accuracy for normal behaviors, 93.81 percent for DoS attacks, 33.67 percent for information scanning attacks, and 99.42 percent for U2R attacks. Wang, J., Li, T., et al. (2010) employed an artificial bee colony algorithm alongside SVM on the KDD Cup '99 dataset, detecting DoS attacks with 99.92 percent accuracy, information search attacks with 100 percent accuracy, U2R attacks with 76 percent accuracy, and R2L attacks with 76 percent accuracy, achieving an overall classification accuracy of 87.92 percent. Mu, Q., Chen, Y., et al. (2012), in contrast, reported 100 percent success in identifying DoS assaults, information scanning attacks, and U2R attacks, along with 99.11 percent success in R2L attacks.

Decision tree are one of the most commonly used classification methods because their training and testing times are short, their findings are simple to read, and they are effective (Diana et al., 2025), Witten, I. H., and Frank, E. (2011). The classification process with decision tree is divided into two parts. The tree is generated in the initial stage. This tree structure is used to generate categorization rules in the second step. In general, the classification procedure is as follows: Let D=t1,t2,..., tn be a database, with each record represented by ti. Let C=C1,C2...Cm denote the collection of classes made up of m classes. Each Cj is a distinct class with its own set of records. That is to say, Cj=ti |=Cj, 1in, and ti D. For each entry in the database, let the fields be A1,A2,...An. Furthermore, if each record belongs to one of the C=C1,C2,...Cm classes, the decision tree can be defined as follows: The Ai field is used to identify each node. Classification rules are the nodes that connect the root node and the leaf node. When generating decision tree, the algorithm utilized is critical. Depending on the algorithm employed, the structure of the tree may alter. Different tree architectures can produce varying categorization outcomes (Ko et al., 2023). Many algorithms have been created that are based on decision tree. These algorithms are classified according to the root, node, and branching criteria. ID3, C4.5, and C5 are well-known algorithms. There are numerous STS research with decision tree in the literature. Table 1 shows the studies that were reviewed. Bahrololum, M., Salahi, E., et al. (2009) obtained classification success of 99.96 percent in identifying normal activities, 99.97 percent in detecting DOS attacks,

99.66 percent in information scanning attacks, 88.33 percent in U2R attacks, and 99.02 percent in R2L attacks using decision tree. They've done it. Furthermore, Alazab, A., Hobbs, M., et al. (2012) used the KDD CUP 99 dataset in 2012 to discriminate between 98.2 percent of normal behaviors, 97.2 percent of DOS assaults, 99.6 percent of information scanning attacks, 92.5 percent of U2R attacks, and 92.5 percent of R2L attacks. They were 99.7 percent successful. Sharma, V., and Nema, A. (2013) conducted another STS investigation utilizing decision tree in 2013 using the KDD CUP 99 dataset. Sharma and Nema detected 99.98 percent of DOS attacks, 88.19 percent of information scanning attacks, 51 percent of U2R attacks, and 94.70 percent of R2L attacks.

Artificial Neural Networks (ANN) are sophisticated categorization tools that mimic the behavior of biological nerve cells (neurons). Each network component is referred to as an artificial nerve (neuron). The neural network is made up of numerous artificial neural cells that are linked together using various weightings. A single neuron can normally only tackle linear issues. Many more categorization issues are solved using multi-layer artificial neural networks (Multi-Layer Perceptron MLP). In function fitting, classification, and matching issues, multilayer ANN is commonly employed. Because of its efficacy in classification, it has been widely employed in IDSs (Bitter et al., 2010). Today's most prevalent Multi-Layer Perceptrons (MLPs) are back propagation networks. It is now one of the most often utilized strategies, particularly in classification procedures. In backpropagation networks, the delta learning rule is utilized as a learning function. Equation 2 expresses the delta learning rule, as seen below.

$$\square_{\square,\square}(\square\square\square) = \square_{\square,\square}(\square\square\square) + (\square * [\square - \square(\square\square\square)] * \square'(\square\square\square)) \quad (2)$$

MLPs are made up of three layers: input, middleware, and output. Information enters the network through the input layer, travels through the intermediate layers to the output layer, and is sent from the output layer to the outside world. Data and results from real-world problems, or examples, are employed in the artificial neural network learning process. The variables linked to the problem form the artificial neural network's input sequence, and the actual outcomes obtained with these variables form the target output sequence that the artificial neural network must achieve (Bengio et al., 2021). Weights are modified during the learning process based on the learning style chosen. Weight change is a metaphor for learning. If there is no weight shift in ANN, the learning process comes to a halt. ANNs have shown excellent outcomes in Intrusion Detection Systems. Cui et al., (2020) constructed IDS utilizing HPCANN (Hierarchical Principal Component Analysis Neural Networks) using the KDD Cup99 dataset; their normal behavior is 97.1 percent, DOS assaults are 100 percent, information scanning attacks are 100 percent, and R2L attacks are 97.2 percent. successfully classified In 2012, Gong, X., and Guan, X. (2012) classified normal behaviors as 100 percent, DOS assaults as 100 percent,

information scanning attacks as 99.76 percent, U2R attacks as 99.85 percent, and R2L attacks as 99.88 percent using the KDD 99 dataset.

Attack and Attack Detection Scenarios. To identify and prevent DDoS attacks on SDN, attack and intrusi

on detection scenarios were built to test the effectiveness of the proposed solution, and the findings were evaluated across these three different scenarios. The findings and discussion section goes into great detail on these instances (Figure 2).

Figure 3 is an example of a DDoS assault scenario. The study employed six OpenFlow Keys. To provide a safe connection, "*A OpenFlow Key*" is connected to "ca *RYU Controller*" and "*B OpenFlow Key*" is connected to "*cb RYU Controller*." The other four OpenFlow switches each have their own networks and users. Some of the users were identified as the DDoS attackers, and the DDoS attackers were then leveraged to develop malware packages to attack the target source.

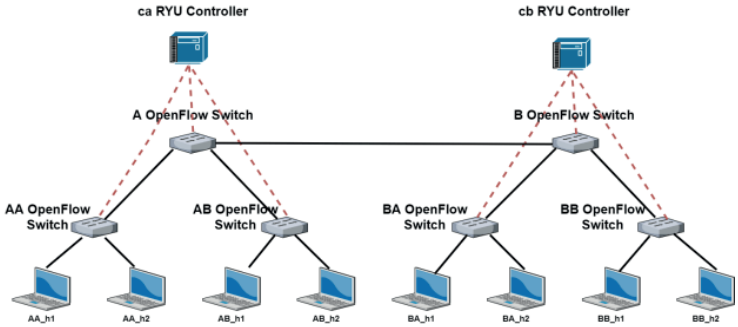

Figure 2 – SDN Topology

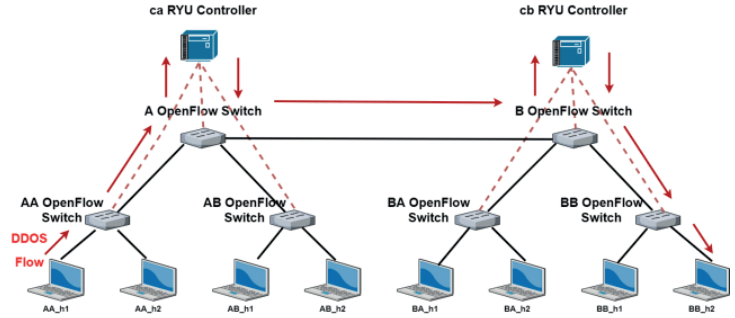

Figure 3 – Sample of DDoS Attack Scenario

The study's primary goal is to detect and prevent DDoS assaults against SDN. Machine learning techniques are used to avoid assaults by quickly developing a high-performance DDoS detection system using information such as the target IP and port number. The premise that every packet arriving at the controller is a new packet is utilized for the destination address parameter, and each incoming packet is evaluated for a valid destination address. If the destination address is invalid or

unknown on the network, the packet is sent to the stream collector. Stream collector is an SDN controller module that stores faulty packets for later analysis. Figure 4 depicts the DDoS detection and blocking system. When the accumulation of invalid packets in a given period becomes considerable, the stream collector sends a notification to the controller, which then constructs a new process for each network device to transfer the invalid packet straight to the stream collector. The stream collector then performs additional processing to the DDoS clustering time model.

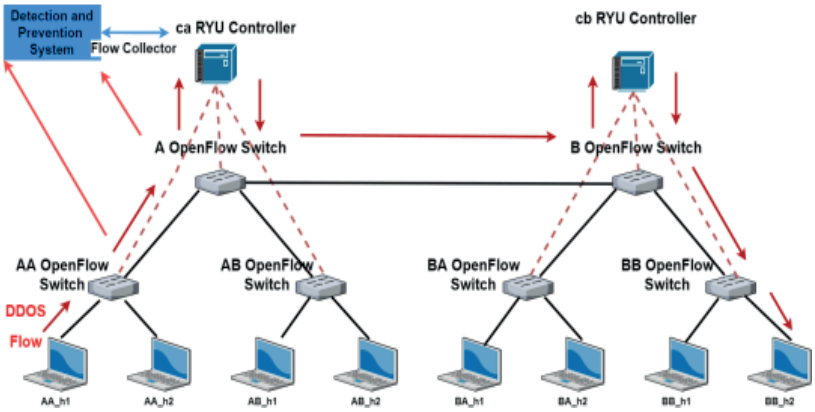

Figure 4 – DdoS Detection and Prevention System

The research universe in the study is all systems that employ SDN because it is connected to the prevention of DDoS attacks that may occur in SDN networks. The representative sample is one in which a person or organization using SDN is subjected to a DDoS attack. Various networking trends influenced the underlying concept of SDN. A properly configured SDN environment can make it easier and more cost-effective to deploy processing capacity to remote sites, move data center functions to the edge, adopt cloud computing, and support IoT settings. The relevant topic of research includes any systems in which such aims are attempted to be realized.

Evaluation Metrics. In the study, the accuracy, precision, sensitivity, and F-measure (F-Score) of algorithms were used to determine the degree of performance of the models developed. Precision and sensitivity alone are insufficient to generate a meaningful comparison result. The F criterion has been defined because assessing both criteria combined yields more accurate results.

Accuracy is defined as the ratio of correctly identified samples ($TP + TN$) to total samples ($TP+TN+FP+FN$).

$$Accuracy = (TP+TN)/(TP+FP+FN+TN) \tag{3}$$

Precision is defined as the ratio of True Positive (TP) samples predicted as class 1 to the total number of class 1 samples anticipated.

$$Precision = TP/(TP+FP) \quad (4)$$

Recall, on the other hand, is a metric that shows how much of the number of True Positive (TP) samples estimated as Class 1 is predicted as Positive.

$$Recall = TP/(TP + FN) \quad (5)$$

The precision value is an important parameter to consider when the cost of estimating as False Negative is large. Because the cost of inaccurate estimation in DDos attacks is considerable, the sensitivity for this study is expected to be as high as possible.

The harmonic mean of precision and sensitivity is the F-scale. We may not be able to draw significant comparative conclusions based solely on precision and sensitivity criteria. When both criteria are evaluated simultaneously, the findings will be more accurate. The F-scale has been defined for this purpose.

$$F\text{-scale}=(2\times Precision\times Precision)/(Sensitivity+Precision) \quad (6)$$

Findings. The findings were evaluated using three alternative scenarios. These;
I. While *AA_h1* and *BB_h1* are communicating, user *BA_h2* starts a DDoS attack against user *AA_h1*,
II. While all users are communicating, user *AA_h1* starts a DDoS attack against user *BB_h1*,
III. While *AA_h1* and *BB_h1* users are communicating, it is the identification and prevention of a DDoS attack from BA_h2 user to *AA_h1*. For this purpose, first

Used SDN Topology. Mininet was used to develop a software defined network topology. Figure 5 depicts the software defined network topology that was constructed.

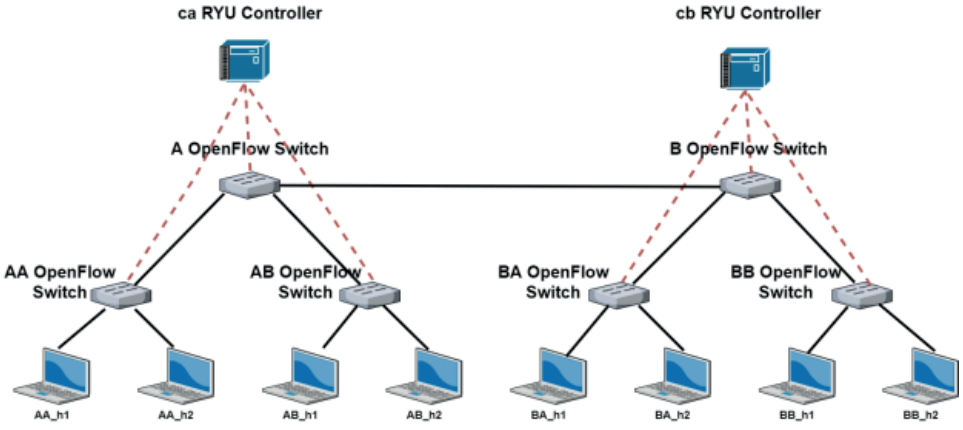

Figure 5 – Created SDN Topology (Created SDN Topology)

In the topology shown in Figure 5, there are two "RYU Controllers," whose names are "ca" and "cb." The "*A Switch*" is linked to the "ca Controller." The "*AA Switch*" and "*AB Switch*" are linked to the "*A Switch*". The "*AA Switch*" is connected to two users. These users are known as "*AA_h1*" and "*AA_h2*". The "*AB Switch*" is connected to two users. These users are known as "*AB_h1*" and "*AB_h2*". The "*B Switch*" is linked to the "cb Controller." The "*BA Switch*" and "*BB Switch*" are linked to the "*B Switch*". The "*BA Switch*" is connected to two users. These users are known as "*BA_h1*" and "*BA_h2*". The "*AB Switch*" is connected to two users. These users are known as "*BB_h1*" and "*BB_h2*."

SDN Normal Traffic Flow. Figure 6 depicts regular network traffic on software defined networks. Pinging BA h1 from user AA h1 was used to test normal flow.

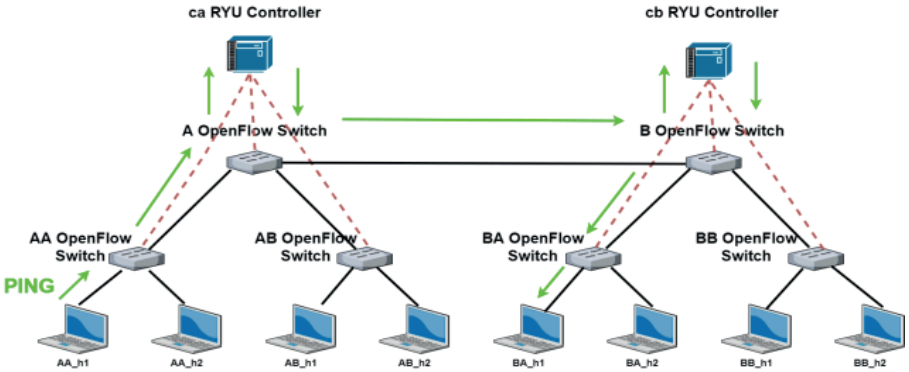

Figure 6 – Normal Traffic Flow in Software Defined Networks

Figure 7 depicts normal traffic resource use on software-defined networks using sFlow.

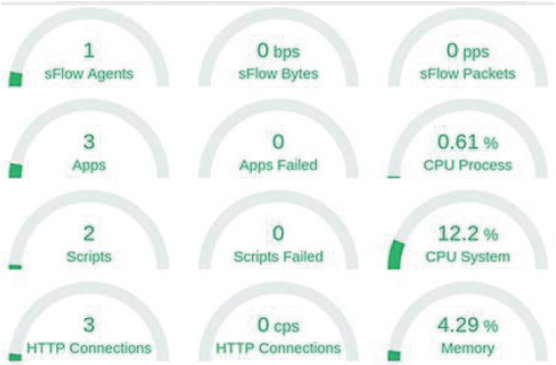

Figure 7 – SDN Normal Traffic Monitored over sFlow

Performing User-to-User DDoS Attack on Software Defined Networks. DDoS refers to the temporary or permanent disruption of a computer's functions. As a result, a DDoS assault is a cyber attack that seeks to prevent genuine users from accessing the computer's network resources (Gangadhar and Sterbenz, 2017).

Different scenarios, such as two DDoS attacks, are presented in this study. These are "DDoS Attack of User *AB h*1 Against User *AA h1* While AA h1 and *BB h1* Communicate" and "DDoS Attack of User *AA h1* Against User *BB h1* While All Users Communicate."

Scenario 1: While user *AA_h1* pings User *BB_h1* during their communication, user *AB_h1* starts a DDoS attack on User *AA_h1* . Figure 8 depicts it. By starting a DDoS attack against user *AA_h1*, user *BA_h2* has depleted *AA_h1*'s resources (Figure 9). The attack has prevented users of *AA_h1* and *BB_h1* from communicating.

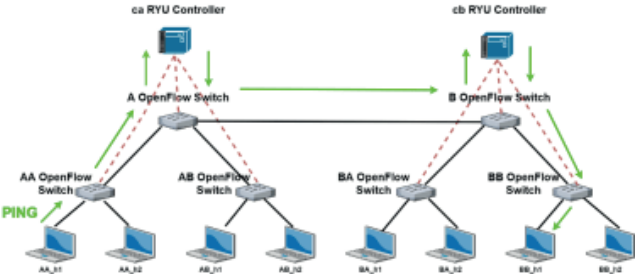

Figure 8 – Normal Communication of *AA_h1* and *BB_h1* Users

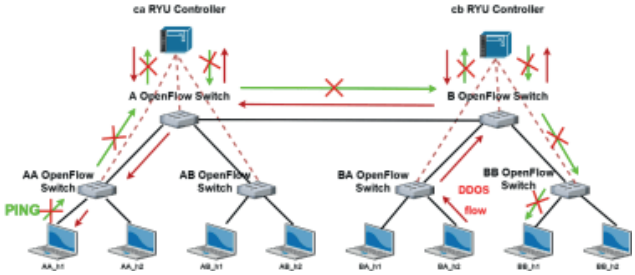

Figure 9 – Failure of *AA_h1* and *BB_h1* Users to Communicate as a Result of DDoS Attack from *BA_h2* Endpoint to User *AA_h1*

Figure 10 – depicts schematically the DDoS assault by user *BA_h2* against user *AA_h1*.

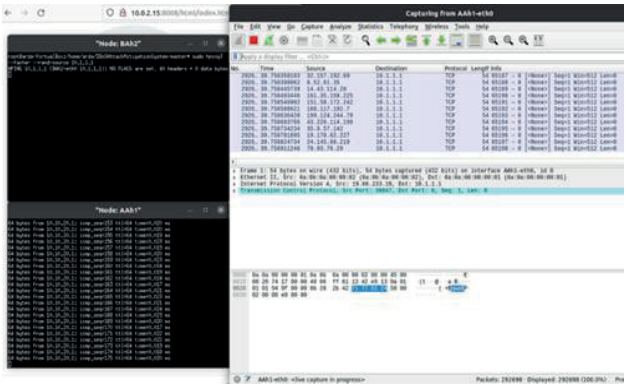

Figure 10 – Attack by user *BA_h2* Against User *AA_h1* Through DdoS

DDoS attacks are displayed using Wireshark. Wireshark was used for packet analysis. DDoS attacks are also depicted visually in Figure 11.

Figure 11 – Wireshark I/O Graph of DDoS Attack of User *BA_h2* to User *AA_h1*

Scenario 2: While all users were communicating, a DDoS attack was started against users *AA h1* and *BB h1*. Ping was used to communicate between network users (Figure 10), and the ping test was monitored on sFlow (Figure 12).

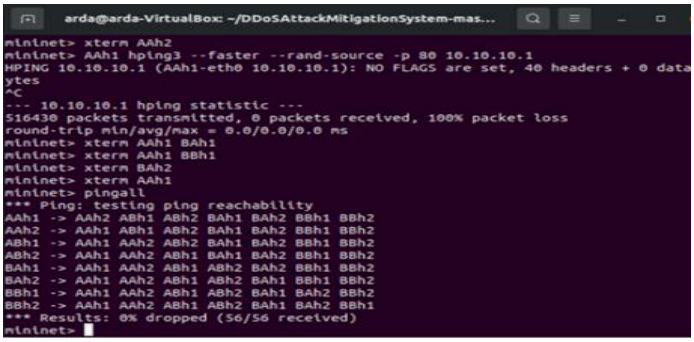

Figure 12 – Communication of Users on the Network

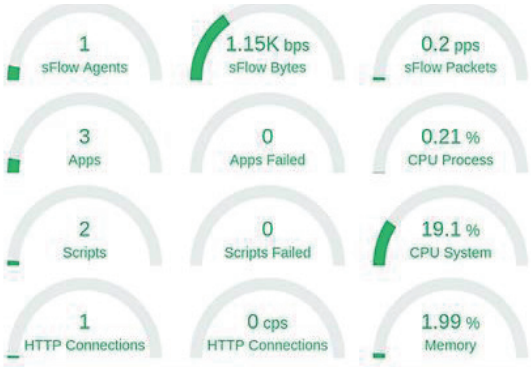

Figure 13 – sFlow Image During Ping Test to Network Users

As seen in Figure 14, a DDoS attack was started by user *AA_h1* against user *BB_h1*.

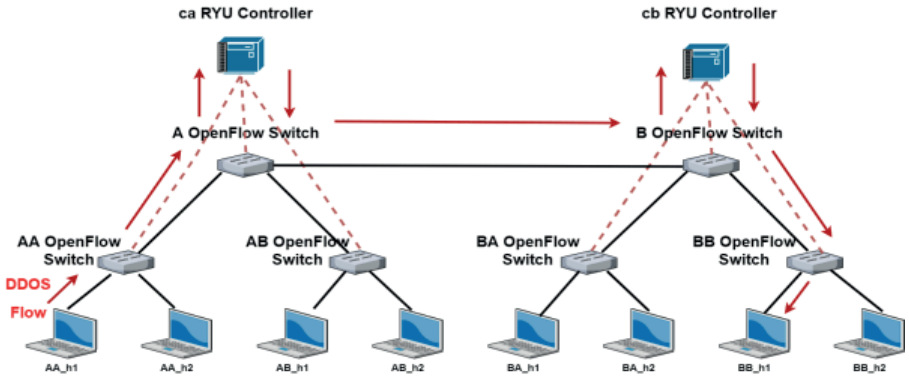

Figure 14 – Starting a DDoS attack on User *BB_h1* by User *AA_h1*

Wireshark was used to perform packet analysis on the DDoS attack output. Figure 14 depicts a DDoS attack that was carried out.

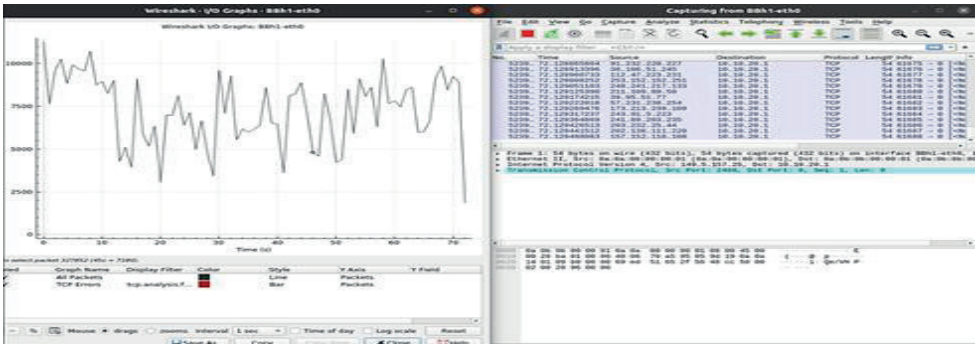

Figure 15 – In Wireshark, user *AA h1* is showing a DDoS attack to user *BB h1*.

As seen in the sFlow graph in Figure 15, DDoS attack consumes the computer's resources.

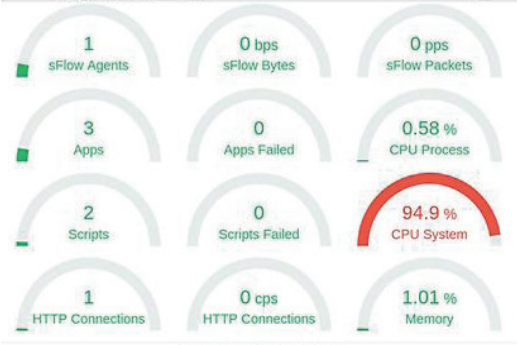

Figure 16 – Showing AA h1's DDoS Attack on BB h1 in sFlow

DDoS Attack Detection in SDN. Machine learning algorithms are used for DDoS attack detection in SDN. The algorithms used were trained and tested using the KDD CUP 99 dataset. For this purpose, KDD CUP 99 dataset was divided into two parts as 70% training dataset and 30% test dataset by hold-out method. A model was created by giving the training data set samples to the Decision Trees machine learning algorithm. This created model has two classes, normal and DDoS. One of these classes shows the normal traffic of the network, while the other class shows a network that has suffered a DDoS attack. All types of DDoS attacks fall under the DDoS class. This classification holds true for Naive Bayes and ANN models as well. Using the KDD CUP 99 dataset, a model was developed using the Naive Bayes machine learning technique. The ANN machine learning technique and the KDD CUP 99 dataset were used to develop a model. However, because the decision tree model has a 91 percent accuracy rate, the decision tree is used.

The Decision Tree model outperformed the ANN (78%) and Naive Bayes models (68%). As a result, the decision tree was selected as the model to be employed in the DDoS detection procedure. The program developed in the study, on the other hand, is structured in a modular structure, allowing DDoS and other intrusion detection systems to be moved to the controller.

Prevention of DDoS Attacks on SDN. After the DDoS assault was identified, the bandwidth was reduced and the computer's network resources were accessible to the principal users.

Scenario 3: While *AA_h1* and BB h1 users were conversing, a DDoS assault from *BA_h2* to *AA_h1* was detected and prevented. User *AA_h1* is pinging user *BB_h1*. User BA_h2 attacked to drain *AA_h1*'s resources by launching a DDoS assault on *AA_h1*. Communication between user *AA_h1* and user *BB_h1* was ensured by recognizing and blocking the assault. Figure 17 depicts it.

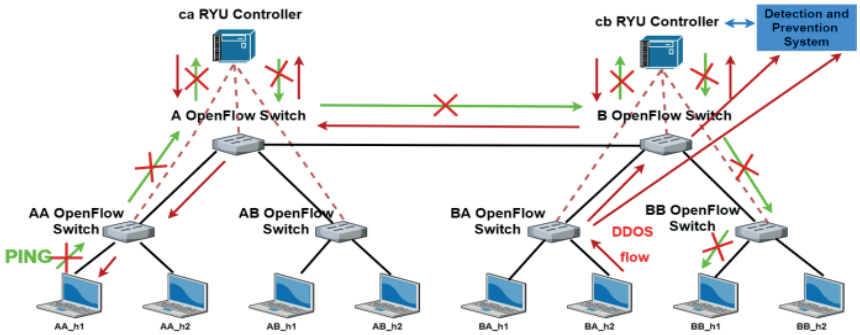

Figure 17 – Ping Test of User *AA_h1* to User *BB_h1* and DDoS Attack Prevention

Results and Discussion. In this work, an artificial intelligence-based application based on SDN was created to detect and prevent DDoS attacks by modeling them on a platform. The security of the network topology is assured by this built application, and precautions are made against vulnerabilities that the configured

network and users may generate. The system was designed in a modular framework for upgrades, enabling the low-cost incorporation of various artificial intelligence models into the system.

The program has been turned into software that requires less human participation in order for the trained models to detect assaults using machine learning methods. In DDoS attacks, a temporary solution is to increase bandwidth so that the user can react to requests. The study mostly detects DDoS attacks. Although shutting the attacked port to avoid DDoS assaults is one of the first solutions that come to mind, controlling bandwidth is recommended since communication cannot be done via the closed port, and so the DDoS attack will succeed. After the DDoS attack was identified, the bandwidth was raised so that the DDoS victim could connect with other users. Table 1 shows the bandwidth that was consumed.

Table 1 – Bandwidth

Bandwidth before DDoS attack	Bandwidth After DDoS Attack	Bandwidth After DDoS Attack Blocked
10bps	4000bps	10bps

According to the predictions, the Decision Tree classifier fared better in terms of the Accuracy measure. However, because making a judgment based on a single statistic can be misleading, the recall and acuity metrics should also be considered. The recall metric value of the Naive Bayes classifier was lower than that of the Decision Tree and ANN. Table 2 displays the metric values. Using the Mininet emulator within a tree network architecture, we conducted a DDoS (Distributed Denial of Service) attack on the Ryu controller (Mehr, S. Y., and Ramamurthy, B., 2019). By employing machine learning technology, specifically support vector machines (SVM), we were able to identify DDoS attacks through the installation of flows in switches. This method allows for the evaluation of the time attack pattern of the DDoS assault, thereby enhancing our detection capabilities. Leveraging our detection technology, we successfully minimized the impact of DDoS assaults on the Ryu controller by 36%. Evaluation of Estimation Algorithms (Test Dataset). The results of tests suggest that it can detect DDoS attacks with a high degree of accuracy.

2 – Evaluation of Prediction Algorithms-Test Dataset

Prediction Algorithms	**Accuracy (%)**	**Recall**	**Precision**	**F_Score**
Decision Tree	91	0.85	0.90	0.87
ANN	78	0.62	0.96	0.75
Naive Bayes	68	0.71	0.95	0.81

Conclusion. The article under review addresses the acute vulnerability of Software-Defined Networks (SDN) to distributed denial-of-service (DDoS) attacks

by proposing a modular, AI-driven detection and prevention framework. It begins by outlining the inherent separation of control and data planes in SDN, which-while simplifying management-exposes the control layer to volumetric and protocol-based DDoS threats.

Using the Mininet emulator, the authors constructed a multi-switch topology managed by dual Ryu controllers to simulate normal and attack traffic flows. They executed three distinct attack scenarios-targeting individual hosts under active communication-to capture packet traces via Wireshark and sFlow. Machine learning classifiers (Decision Tree, Artificial Neural Network, and Naive Bayes) were trained on the KDD Cup ’99 dataset (70% training, 30% testing) to distinguish between normal and DDoS traffic.

The modular architecture-allowing easy reintegration of improved models trained off-line on platforms like Google Colaboratory-stands out as a practical strength, potentially reducing retraining overhead. However, reliance on the outdated KDD Cup ’99 dataset may limit adaptability to evolving DDoS tactics. Experimental validation remains confined to emulated topologies; real-world deployment on production SDN controllers (e.g., OpenDaylight) would better gauge performance overhead and scalability.

This proof-of-concept demonstrates that a Decision Tree–based SI-SDN (Security Intelligence for SDN) module can detect and mitigate DDoS with high accuracy and low latency. Future work should employ contemporary, diversified traffic datasets (e.g., CIC-IDS2017), extend evaluations to hybrid cloud-edge environments, and assess integration overhead on commercial SDN controllers. Such enhancements will solidify the framework’s readiness for operational network defenses.

References

Alazab A., Hobbs M., Abawajy J., and Alazab M. (2012) Using feature selection for intrusion detection system. In International Symposium on Communications and Information Technologies (ISCIT). — P. 296–301. IEEE. https://doi.org/10.1109/ISCIT.2012.6380910 (in Eng.).

Alhakami W., Alharbi A., Bourouis S., Alroobaea R., and Bouguila N. (2019) Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. IEEE Access, 7. — P. 52181–52190. https://doi.org/10.1109/ACCESS.2019.2912115 (in Eng.)

Bahrololum M., Salahi E., and Khalegni M. (2009) Machine learning techniques for feature reduction in intrusion detection systems: A comparison. In Fourth International Conference on Computer Sciences and Convergence Information Technology. IEEE. https://doi.org/10.1109/ICCIT.2009.89 (in Eng.)

Bengio Y., LeCun Y., and Hinton G. (2021) Deep learning for AI. Communications of the ACM, 64(7). — P. 58–65. https://doi.org/10.1145/3448250 (in Eng.)

Bitter C., Elizondo D.A., and Watson T. (2010) Application of artificial neural networks and related techniques to intrusion detection. In International Joint Conference on Neural Networks (IJCNN). IEEE. https://doi.org/10.1109/IJCNN.2010.5596532 (in Eng.)

Çalış K., Gazdağı O., and Yıldız O. (2013) Reklam içerikli epostaların metin madenciliği yöntemleri ile otomatik tespiti [Automatic detection of advertising e-mails using text mining methods]. Bilişim Teknolojileri Dergisi, 6(1). (in Turkish).

Cui Y., Jin Z., and Hu J. (2020) Research on intrusion detection method based on hierarchical

self-convergence PCA-OCSVM algorithm. International Journal of Network Security, 22(6). — P. 916–924. https://doi.org/10.6633/IJNS.202011_22(6).04 (in Eng.)

Diana L., Dini P., and Paolini D. (2025). Overview on intrusion detection systems for computers networking security. Computers, 14(3), Article 87. https://doi.org/10.3390/computers14030087 (in Eng.)

Gangadhar S., and Sterbenz J.P.G. (2017) Machine learning aided traffic tolerance to improve resilience for software defined networks. In 2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM). — P. 1–7. IEEE. https://doi.org/10.1109/RNDM.2017.8093035 (in Eng.)

Gong X., and Guan X. (2012) Intrusion detection model based on the improved neural network and expert system. In IEEE Symposium on Electrical and Electronics Engineering (EEESYM). IEEE. https://doi.org/10.1109/EEESym.2012.6258621 (in Eng.)

Huang W., Li G.-M., and Chen W.-W. (2018) A review of statistical learning theory. DEStech Transactions on Engineering and Technology Research. https://doi.org/10.12783/DTETR/PMSMS2018/24953 (in Eng.)

Ko K., Baek J., Seo B.-S., and Lee W.-B. (2023) Comparative study of AI-enabled DDoS detection technologies in SDN. Applied Sciences, 13(17), Article 9488. https://doi.org/10.3390/app13179488 (in Eng.)

Mehr S.Y., and Ramamurthy B. (2019) An SVM-based DDoS attack detection method for Ryu SDN controller. In Proceedings of the 15th International Conference on Emerging Networking EXperiments and Technologies. — P. 72–73. ACM. https://doi.org/10.1145/3360468.3368183 (in Eng.)

Mu, Q., Chen, Y., and Zhang, Y. (2012). Incremental SVM algorithm to intrusion detection based on boundary areas. In International Conference on Systems and Informatics. IEEE. https://doi.org/10.1109/ICSAI.2012.6223447 (in Eng.)

Panigrahi, A., and Patra, M. R. (2019). Anomaly based network intrusion detection using Bayes net classifiers. International Journal of Scientific and Technology Research, 8(9). — P. 481–485. https://www.ijstr.org/paper-references.php?ref=IJSTR-0919-21989 (in Eng.)

Peng H., Sun Z., Zhao X., Tan S., and Sun Z. (2018) A detection method for anomaly flow in software-defined network. IEEE Access, 6. — P. 27809–27817. https://doi.org/10.1109/ACCESS.2018.2839684 (in Eng.)

Raktate G., Shelar K., Parjane P., Pangavhane S., More S., and Deshmukh S. (2024) A survey on security issues and challenges in cloud computing. — P. 1–5. https://doi.org/10.1109/DASA63652.2024.10836628 (in Eng.)

Ray D. (2024) Internet traffic classification using machine learning techniques. https://doi.org/10.55041/isjem01429 (in Eng.)

Sharma V., and Nema A. (2013) Innovative genetic approach for intrusion detection by using decision tree. In International Conference on Communication Systems and Network Technologies (CSNT). IEEE. https://doi.org/10.1109/CSNT.2013.93 (in Eng.)

Sultana N., Chilamkurti N., Peng W., and Alhadad R. (2019) Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications, 12(2). — P. 493–501. https://doi.org/10.1007/s12083-017-0630-0 (in Eng.)

Wang J., Li T., and Ren R. (2010) A real time IDSs based on artificial bee colony support vector machine algorithm. In Third International Workshop on Advanced Computational Intelligence. IEEE. https://doi.org/10.1109/IWACI.2010.5585107 (in Eng.)

Witten I.H., and Frank E. (2011) Data mining: Practical machine learning tools and techniques (3rd ed.). Morgan Kaufmann. (in Eng.)

Zhang Y., and Zhu Y. (2010) Application of improved support vector machines in intrusion detection. In 2nd International Conference on e-Business and Information System Security. IEEE. https://doi.org/10.1109/EBISS.2010.5473653 (in Eng.)

Publication Ethics and Publication Malpractice
in the journals of the Central Asian Academic Research Center LLP

For information on Ethics in publishing and Ethical guidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the journals of the Central Asian Academic Research Center LLP implies that the described work has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The Central Asian Academic Research Center LLP follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the Cross Check originality detection service http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will only accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the Central Asian Academic Research Center LLP.

The Editorial Board of the Central Asian Academic Research Center LLP will monitor and safeguard publishing ethics.

Правила оформления статьи для публикации в журнале смотреть на сайтах:
www:nauka-nanrk.kz
http://physics-mathematics.kz/index.php/en/archive
ISSN2518-1726 (Online),
ISSN 1991-346X (Print)

Ответственный редактор *А. Ботанқызы*
Редакторы: *Д.С. Аленов, Т. Апендиев*
Верстка на компьютере: *Г.Д. Жадырановой*

Подписано в печать 31.03.2026.
Формат 60x881/8.
20,0 п.л. Заказ 1.

«Central Asian Academic Research Center» LLP
Алматы, Шевченко к-сі, 28