JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An adaptive approach to Detect and Mitigate DDOS attack using machine learning

Assistant Professor Dr. K Vanitha, Akash Balasubramani, Akanksha, Jayasurya P Gowda, Gautham N T Jain University, Computer Science Department, Bangalore, India.

Abstract–The future of networking is in software defined networks, which allow for centralized network control by separating network devices' data and control plane. SDN can deliver superior administration and security of a network and enables us to program the network for improved speed and usability. DDOS attacks are the most severe and threatening attacks in a network. They can flood the network, block access to the server network with enormous packets, and exploit network resources to prevent answers for future incoming requests. Nonetheless, SDN is subject to attacks. DDOS attacks are known to only become more common in a cloud setting. To effectively detect and mitigate DDOS assaults in SDN, a method for doing so is described that combines statistics and machine learning techniques. The machine learning technique used in implementing this method has achieved an accuracy of 99.26% and a detection rate of 100% in detecting and mitigating DDOS assaults in a software defined network utilizing the Ryu controller and Mininet network simulator with OpenFlow SDN protocol.

I. INTRODUCTION

Because of the crucial advantages it may provide over traditional networks, the trend of cloud computing has rapidly expanded in recent years in both the industrial and academic sectors. In the field of networking and popular cloud networks, software defined networks (SDN) have experienced tremendous growth. SDN is a networking technology that enhances network management and performance. It allows for centralized network management and the ability to program network devices.

By separating the data and control planes of the network device, software defined networking makes it possible for an SDN controller to program the control plane. Three layers comprise SDN architecture: the application layer for networking applications, the control layer where the controller is implemented, and the infrastructure layer where networking devices like switches and hosts are present.

SDN monitors and operates the network from a single location, making it easier to change and configure network devices. It enhances flexibility, cloud management, scalability, performance, and controllability. To have better security and control over the network and deliver networking as a service, software defined network-based cloud environments have been used in cloud computing networks.

II. CONTRIBUTION

In this paper, a strategy for combining statistical analysis and machine learning techniques is given. Four features are taken from network traffic and compiled into a dataset using statistical methods. This dataset is used to train the support vector machine classifier machine learning algorithm to predict DDOS attacks in the network and detect malicious traffic early and mitigate it by blocking the source and port in the network.

The implemented method was evaluated and found to be 99.26% accurate in anticipating the attack. Moreover, 100% of the malicious traffic was found, and there were zero false alarms from network traffic.

III. Related Work

These sections comprise this report: The work carried out by other researchers on this topic is described in Section 2: Relevant Work. The methodology offered in this study/work is shown in Section 3, along with the tools and technologies employed. The SDN framework and the network topology are built for simulation as described in Section 4 of the design specification. Section 5 describes the implementation methods and the software and tools used in this project. The evaluations, configurations, and experiments in Section 6 show the outcomes and how the project functions. The project's completion and future efforts covered in Section 7. The review of the literature splits into three parts: Traffic analysis in SDN,

SDN and machine learning technologies for anomaly detection are used for DDOS attack detection and mitigation. These parts offer an extensive overview of the work done by various researchers in the disciplines and approaches in combating DDOS attack utilizing SDN.

Traffic Analysis In SDN

The methods and work presented and implemented by researchers in analyzing incoming traffic in a software defined network are described in this section. DDOS attacks and traffic analysis can be detected using the following two techniques:

• Network data analysis

• Data SNMP analysis

Data analysis on the network To determine the possibility of a DDOS attack, this method involves identifying malicious traffic in the network. Every network traffic has a specific set of features, and it is necessary to extract those features by specifying a few aspects of the attack flow. To distinguish between normal traffic and attack traffic, it is also necessary to record the behavior and characteristics of normal traffic. Many

researchers have suggested various methods and feature extraction parameters. swiftly identify possible victims, malicious traffic, and suspicious attackers, the author Xu and Liu (2016) have presented a concurrent algorithmic method that modifies the monitoring capabilities of the flow on the switches in the network. The writers have tested the designed method, and the graphs and findings in the article strongly support the author's claims. The results were acquired with great accuracy by capturing asymmetric flow; however, the report does not go into enough depth about the tools and modeling techniques used to get there.

By collecting traffic data and counting the byte rate, symmetric and asymmetric flow changes, and small amounts of incoming packets in the network, Wang et al. (2019) have presented four feature extraction approaches. The results reveal that the controller reaction time under a DDOS attack has decreased. The proposed algorithm is implemented using the Ryu controller and simulated using the Mininet. He et al. have used a similar strategy but with different feature extraction parameters and a density peak clustering algorithm (2017). Before applying the density peak clustering algorithm to the correlated data for the network's DDOS attack detection, anomaly detection is first carried out by gathering traffic features and then determining correlations with the characteristics of malicious traffic for both solid and weak correlation factors. The authors used the MINE package and a few open-source Python machine-learning libraries to implement the algorithms in Python 2.7.9. Although the authors assert their method outperforms existing ML algorithms, it does not consider real-time information about incoming network traffic.

In this paper, Jin et al. (2003) have proposed a protection system against DDOS faked traffic by employing a hop-count filtering method; every IP packet has to take a specific number of hops to reach its destination. The authors claim that their method, which counts the number of hops and the TTL (Time to Live) value in the IP header, can detect 90% of spoof IP packets. The investigations were carried out by creating TCP and ICMP traffic using the Linux kernel. Graphs effectively illustrate the results, but the experimental setting is time- and labor-intensive.

DDOS Attack Detection and Mitigation Using SDN

This section contains related research on DDOS attack detection and mitigation in SDN utilizing conventional techniques and cutting-edge detection methods developed by other researchers.

Cui et al. (2016) provided a software defined-anti DDOS defense mechanism solution. The objective is to have an attack detection trigger mechanism with an attack trackback system. The detection strategy is based on related research from other researchers. The model employs four stages: initial, detection, traceback, and mitigation. The innovative work given determines the number of records from the traffic. It calculates the velocity of the packets to detect malicious traffic in the network. It also traces the source IP from where the DDOS assault is coming. The simulation environment is built using Mininet, and the approach is implemented using the Python-based Ryu controller. The author's assertions are supported by well-documented results and by evidence that can be used to repeat the experiments that were carried out.

While most DDOS attacks fill up the flow table entries and block new entries in the switch, Bhushan and Gupta's (2019) solution is to provide a mitigation mechanism for DDOS attacks by configuring the size of the flow table in the software defined network.

We store Blacklisted sources and flow table status in two databases in the design. The blocklist status maintains the source IPs of the malicious attack traffic coming into the network. In contrast, the flow table status holds the flow entries currently in the network from all switches. When the flow table is complete, it looks for the closest switch to reroute the traffic and prevent an obstruction due to flow table size. This method permits the malicious traffic in the network, which might be dangerous to the network and does not have an early detection scheme. The authors tested the method and then presented their findings. Pox Controller, a Python-based controller, was used for implementation, and Mininet was used to build the simulation environment. We can replicate and duplicate the experiment using the evaluations. The only restriction is the lack of an early detection approach.

Machine Learning Approaches for Attack Detection

This section describes the many machine learning approaches and algorithms that numerous researchers have employed to forecast and identify DDOS traffic in an SDN. Santos et al. (2019) have examined the many machine learning techniques that may be utilized with software defined networks to identify and reduce DDOS traffic in a network in this paper.

The authors show the implementation of four machine learning (ML) algorithms—MLP, Decision Tree, Support Vector Machine (SVM), and Random Forest. We stimulate all ML algorithms using Mininet. According to the results, decision trees and random forest algorithms perform best for DDOS attack detection in terms of accuracy and processing speed. Unfortunately, there were a few issues with implementing the bandwidth attack and flow table attack. The post is comprehensive and well-written overall.

Research conducted by Sahoo et al. (2018) compared several machine learning methods, including k-nearest neighbor (KNN), naive Bayes (NB), support vector machine (SVM), random forest, and linear regression (LR). The findings of the experiment demonstrate that the best prediction accuracy of 98% was achieved by the linear regression (LR) and random forest machine learning algorithms, with random forest taking less time to execute than LR. Although the article has the result displayed, the tools used for implementation and simulation are not described in detail, making it impossible to duplicate this work.

We have seen the support vector machine ML approach, one example of how machine learning algorithms constantly change and improve at making predictions. However, Myint Oo and colleagues suggest a cutting-edge ML technique based on support vector machines. The ASVM algorithm is used in this study to gather data from the feature extraction stage and categorize parameters to anticipate DDOS assaults in SDN. This method is said to shorten the testing and training periods required for the machine learning algorithm to complete its tasks. The authors claim that the ASVM approach has a detection accuracy of 97% with the quickest testing and training times. We have implemented this using an Opendaylight controller and constructed the simulation environment using a Mininet. The paper's graphical results provide strong evidence supporting the author's claims.

In order to improve forecast accuracy for DDOS attacks, Mohammed e al. (2018) combined traditional techniques with machine learning by training the model with the NSL-KDD dataset. Four SDN controllers—POX, ONOS, Ryu, and Opendaylight—are used in the experimental setup. In contrast, Mininet is used to simulate the network. Li et al. (2018) suggested a similar strategy but with deep learning in OpenFlow based SDN, wherein traditional neural networks models like CNN, RNN, and LSTM are applied with deep learning to detect incoming malicious DDOS traffic better detect incoming malicious DDOS traffic.

IV. Conclusion

Researchers analyzing incoming traffic in software defined networks have demonstrated that each network flow has specific parameters and features that may be tracked and gathered to extract the precise attributes needed to identify malicious DDOS traffic in a network. Various techniques given by researchers ensure improved security against DDOS assaults using their innovative methods and implementation.

Although there are many ML approaches, the related work utilizing ML methods reveals that it can achieve improved accuracy in detecting anomalies in the network traffic. Applying machine learning algorithms to detect anomalies in the network is still a prominent research topic. There are some instances where the implementation techniques used in related work vary. Nonetheless, Mininet is the most widely used and best option for building the modeling environment for software defined networks. Ryu Controller is one of the well-known Python-based Programmable Controllers.

V. Methodology

Traditional network systems are incredibly vulnerable to attacks, which can result in data leaks and privacy concerns. This work provides a software defined network-based DDOS attack detection and mitigation strategy combining statistical network analysis and machine learning techniques to prevent sucky network attacks on the public network. An SDN-based network can separate the control plane and data plane from the network devices. We set up a centralized management system to guard against illegal access to the network. Every network packet flow that enters the network has a few attributes and characteristics set forth; these characterizations are gathered as training and test features for our method of DDOS attack prevention on the network using software defined networking. To identify DDoS assaults, the following characteristics and parameters are tracked and gathered:

1. Speed of IP Sources: This feature displays the total number of incoming TP sources in the network during a specified period of time. SSIP, the acronym for it, is defined as.

$$SSIP = Sum\ IPsrc/T$$

Where, T is the sample time intervals, and Sum IPsrc is the total number of IP sources entering each flow. The detection system monitors flows, gathers data on them every three seconds, and stores the number of source IPs during this time. This is accomplished by setting the time interval T to three seconds. For the machine learning system to predict attacks, the controller must have enough data on both regular and attack traffic. The SSIP for typical attacks is typically low, whereas the attack count is generally greater.

2. Traffic Flow Count: Each network packet that enters the network has a specific amount of flow counts. DDOS attack traffic has a lower flow count than regular traffic.

3. Speed of Flow Entries: The total number of flow entries to the network switch within a specific period. It is known as SFE and is described as;

$$SFE = N / T$$

This is a fundamental characteristic of attack traffic identification because, in contrast to the speed of flow entries value of typical traffic flows, the number of flow entries for DDOS attacks significantly increases over the course of a set period.

4. Pair-Flow Entries Ratio: This ratio is the sum of all incoming flow entries into the switch, which are interactive IPs, divided by the sum of all flows throughout the T time period. RPF is referred to as and is understood to be;

$$RPF = ScrIPs / N$$

where N is the total number of IPs and ScrIPs is the total number of collaborative IPs in the network flow. Under typical traffic conditions, the IP source of the ith flow will be the same as the IP of the jth flow's destination, and the source of the jth flow will be the same as the IP of the ith flow's destination. , a DDoS assault traffic, it will not be an interactive flow in this scenario. The host destination is under attack when flow entries to it increase quickly at time T, and the host there is unable to respond to them.

As a result, as soon as the DDOS attack begins, the overall number of collaborative flows in the attack traffic will abruptly fall. The total number of collaborative flows is divided by the entire number of flows in order to broaden the network's use of this detection parameter under various operational scenarios.

These four parameters and distinguishing characteristics are taken from each incoming traffic flow and are set in the SDN Ryu controller. The SVM/Decision tree machine learning system is taught to detect malicious traffic entering the network and classify it as either regular or DDOS traffic using these extracted feature data.

Machine learning Algorithm

The Support Vector Machine (SVM) classifier is a supervised machine learning algorithm that distinguishes between hyperplanes. SVM learns from the data that is given to it and that it has been trained with, comparing the data that has been assigned to it with the data that was used to train the algorithm.

It develops a pattern map that separates the characteristics of regular traffic from those of attack traffic.

The distinction between decision tree classifier and SVM is in how the latter analyses and categorizes data. Decision tree classifier is also a machine learning method. The data is divided into smaller pieces using

the decision tree method, and the outcome is obtained as a tree structure. For the SVM and decision tree classifier algorithms, Python includes built-in libraries. Both machine learning algorithms may be used withthe controller thanks to programming. However, the support vector machine (SVM) ML technique is used to test and experiment with the proposed method.

Development and simulation Platform Tools

In this project, the method is put into practice in a virtual setting made using VMware Workstation. Ubuntu 20.04 is set up in Virtual Machine to create the simulation's operating environment. The proposed methodology was put into practice using the next set of tools and technology.

In a software defined network, the forwarding plane of a network switch or router can be accessed via the OpenFlow communication protocol.

We create a virtual network of hosts, switches, controllers, and links by the network emulator known as Mininet. Standard Linux network software is used by Mininet hosts, and its switches support OpenFlow for Software-Defined Networking and very flexible custom routing. Mininet is an open-source network simulator for software-defined networks that can mimic a huge network in a virtual setting. The main benefit of using the Mininet is that it supports OpenFlow Protocol, which is necessary for Software Defined Network design and processing.

Additionally, it offers a cheap platform for designing, testing and constructing unique network topologies. The Ryu Controller is an open-source, software-defined networking (SDN) controller created to improve network agility by simplifying the management and adaptation of traffic management. It is a Python-based programmable controller tool.

For measuring network performance, we use the IPERF tool here that is used to gauge a network's bandwidth and datagram loss. The network throughput and data streams for the User Datagram Protocol (UDP) and Transport Control Protocol (TCP) are measured in this study. The iperf utility creates client and server functionality for both the source and destination nodes to aid in measuring network performance.

VI. Design Specification

The SDN framework that is being described features a data plane with numerous nodes/hosts that are virtually constructed using Mininet and coupled to an OpenFlow switch that defines the SDN protocols and communicates with the framework's control plane via the OpenFlow protocol.

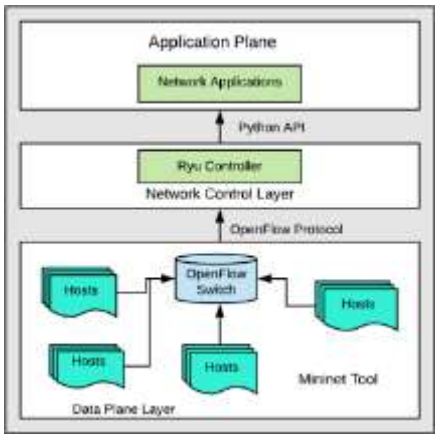


Figure 2: SDN Framework.

In this case, the Ryu controller is used as the controller, which gives the programming capabilities and allows us to control the routing activities in the control plane, which controls the data plane and the switches, defines rules, and also monitors the network traffic flow. Since Ryu is a Python-based controller that communicates with the application layer in this example, network traffic applications using a Python-based API, the control plane is programmed in Python.

1.1.Network Design

Figure 3: Mininet Network Design.

The network topology was created using the Mininet network simulator, and the network consists of 25 hosts/nodes, one OpenFlow switch, and one Ryu controller. The switch is connected to all of the hosts, and the controller is connected to the switch. All of these hosts and switches are under the authority of the Ryu controller, and any port that is being attacked will be promptly stopped.

VII. Implementation

In order to identify and counteract DDOS attacks in a software defined network, the proposed strategy combines statistics and machine learning techniques. The implemented technique calls for training the SVM ML algorithm to recognize network attacks.

The Data Collection module must gather information on both legitimate and malicious traffic, then save it in a CSV file for the ML algorithm to use. Prior to collecting attack traffic data, standard traffic data must first be gathered. For greater accuracy, it is advised to collect standard traffic data once more after ordering attack traffic data. The three statistical feature extraction parameters specified in the methodology—speed of source IP, speed of flow entries, and ratio of flowpair entries in different columns—reconsidering when collecting the data.

When the normal traffic is generated, the SVM algorithm predicts that it is normal traffic, and when the attack traffic is generated, it immediately recognizes the traffic as DDOS attack traffic and blocks the port from which the traffic is incoming. Detection and mitigation take place after the data has been collected and the controller is set to detection state. The controller is configured to a 120 second hard time once the port has been blocked, after which the port is unblocked. If

the attack is still going on, however, it once more identifies it and blocks the port for an additional 120 seconds. Following blocking, other ports on the network are free to receive regular traffic. While the attack is ongoing, this process continues.

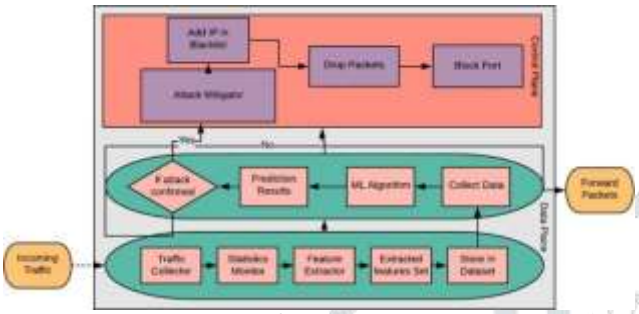


Figure 4: Flowchart of the Presented Method.

Each protocol and controller used in software defined networks is created to execute a specific function and offer efficiency and flexibility in a certain area. The most well-liked and effective tools for DDOS attack detection and mitigation in a software defined network are used to achieve the suggested strategy.

OpenFlow Protocol is the most widely used and accepted protocol for software defined networks and hence Open V-switch is used in this project. The reasoning and approaches are programmed in Python since the provided solution combines statistics and machine learning methods. The logic for these parameters is all built into the controller and includes parameters like the speed of the source IP, the speed of the flow entries, and the ratio of flowpair entries.

A programmable controller called **Ryu Controller**, which is open-source and based on Python, is used to specify the guidelines and logic that the switches in the technique should adhere to.

In this work, a single openVswitch with 10 and 25 hosts is generated for various experiments. **Mininet** is a network simulator that produces a virtual network architecture including controller, switches, and hosts.

A packet generator called **Hping3** is mostly used to assess network security since it generates TCP/IP traffic in the network. Using this tool, programs are created to automatically generate both normal and attack traffic.

Iperf, which is utilized in this work to manually produce traffic, is also a network traffic generator and performance tester. The Ubuntu 20.04.1 LTS operating system, on which VMware Workstation is installed, has all of these tools pre-installed.

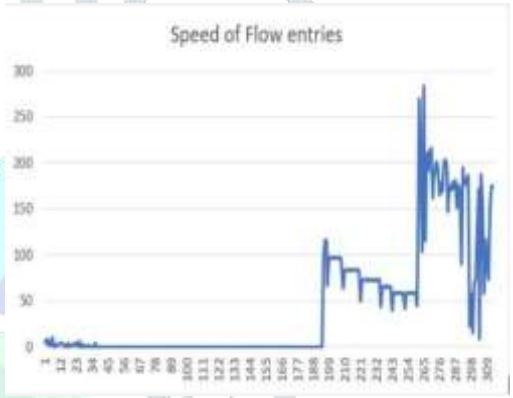
Evaluation

This section describes the tests performed on the SDN using legitimate and malicious traffic sent to the network from various ports, as well as the general process of detection and mitigation, including the precision and detection rate of the adopted approach, For the SVM algorithm to train on and conduct analyses of the attack, the datasets were initially built with 600+ samples of standard traffic data and 300+ samples of attack traffic data. SVM predicts the traffic every 2 seconds during the tests, which last 300 seconds and include traffic collection intervals of 2 second

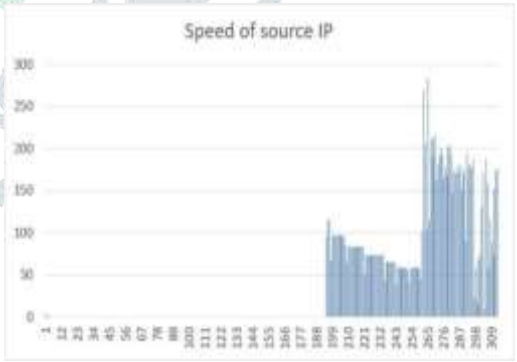
Experiment / Case Study 1

In this experiment, all ports send regular traffic, while port/host 1 in the network sends an attack, with incoming traffic collected every three seconds. Mininet is used to build the network topology, which consists of 10 hosts and 1 openflow switch.

The X-axis in graphs A and B represents data counts, while the Y-axis represents the speed count of flow entries and source IP. The graph shows how to attack traffic causes the speed of flow entries and the rate of IP sources to increase, while the straight line represents regular traffic flow in the network.



(a) SFE



(b) SSP

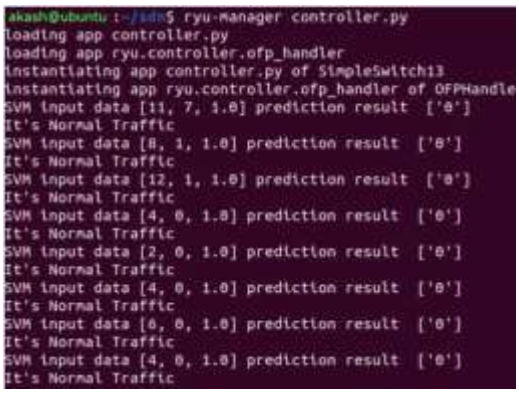


Figure 7: Normal Traffic Prediction

SVM machine learning algorithm predicting the traffic as normal traffic.

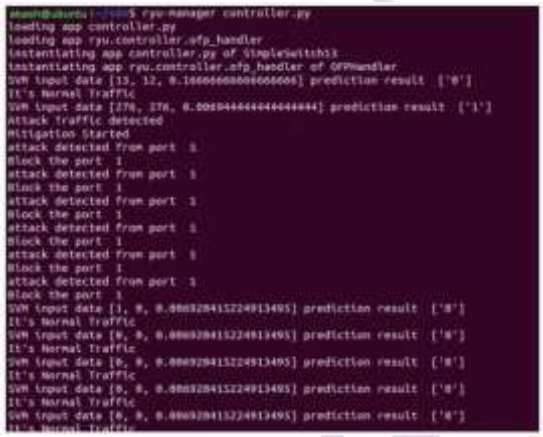


Figure 8: Attack Traffic Prediction

The SVM machine learning algorithm recognizes the traffic as DDOS attack traffic, initiates an immediate mitigation step, and blocks port 1 where the attack traffic is coming from.

Figure 10: Accuracy Score

The network's DDOS attack traffic detection percentage using the proposed method is 92.8%, with 0% false alarms, implying that no legitimate traffic was ever mistaken for an attack.

VIII. CONCLUSIONS

In contrast to traditional networks, software defined networks allow us to create and manage network operations through programming. The principal objective of this effort was to use SDN to identify and counteract DDOS attacks in a cloud setting. To detect and predict DDOS attacks in the network, we have implemented a method that combines statistical features like the source of the IP address, the speed of the flow entries, the flowcount, the ratio of the flow-pair, and the SVM machine learning algorithm. Experiments have shown that the method can provide an accuracy of 99.26% and a detection rate of 100% for malicious traffic with 0 false traffic predictions.

IX. REFERENCES

1) Zaher M., Alawadi A.H. Sieve: A flow scheduling framework in SDN based data center networks. *Comput. Commun.* 2021;171:99–111. doi: 10.1016/j.comcom.2021.02.013.
2) Liu G.Y., Guo S. SDN-Based Traffic Matrix Estimation in Data Center Network through Large Size Flow Identification. *IEEE Trans. Cloud Comput.* 2022;10:675–690. doi: 10.1109/TCC.2019.2944823.
3) Fogli M., Giannelli C. Software-Defined Networking in wireless ad hoc scenarios:Objectives and control architectures. *J. Netw. Comput. Appl.* 2022;203:103387. doi: 10.1016/j.jnca.2022.103387.
4) Aslam M., Ye D., Hanif M., Asad M. Machine learning based SDN-enabled distributed denial-of-services attacks detection and mitigation system for Internet of Things; Proceedings of the International Conference on Machine Learning for Cyber Security; Guangzhou, China. 8–10 October 2020; pp. 180–194.
5) Polat H., Turkoglu M., Polat O. A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks. *Expert Syst.Appl.* 2022;197:116748. doi: 10.1016/j.eswa.2022.116748.
6) Peng J.C., Cui Y.H., Qian Q. ADVICE: Towards adaptive scheduling for data collection and DDoS detection in SDN. *J. Inf. Secure Appl.* 2021;63:103017. doi: 10.1016/j.jisa.2021.103017.
7) Soylu M., Cuillen L., Izumi S. NFV-GUARD: Mitigating Flow Table-Overflow Attacks in SDN Using NFV; Proceedings of the IEEE 7th International Conference on Network Softwarization; Tokyo Japan. June 28–July 2 2021.
8) Segura G.A.N., Chorti A. Centralized and Distributed Intrusion Detection for Resource-Constrained Wireless SDN Networks. *IEEE Internet Things J.* 2022;9:7746–7758. doi: 10.1109/JIOT.2021.3114270.
9) Agrawal N., Tapaswi S. An SDN-Assisted Defense Mechanism for the Shrew DDoS Attack in a Cloud Computing Environment. *J. Netw. Syst. Manag.* 2021;29:12. doi: 10.1007/s10922-020-09580-7.
10) Shah S.Q.A., Khan F.Z. Mitigating TCP SYN flooding based EDOS attack in cloud computing environment binomial distribution in SDN. *Comput. Commun.* 2022;182:198–211. doi: 10.1016/j.comcom.2021.11.008.