Computer Assisted Methods in Engineering and Science
30(2): 167–186, 2023, doi: 10.24423/cames.450

This article belongs to the *Special Issue on Scientific Computing and Learning Analytics for Smart Healthcare Systems* edited by Dr. C. Chakraborty, Dr. S. Barbosa and Dr. L. Garg

Detection of Distributed Denial of Service Attacks for IoT-Based Healthcare Systems

Gaganjot KAUR*, Prinima GUPTA

Department of Computer Science and Technology, Manav Rachna University, Faridabad, India, e-mail: prinima@mru.edu.in

* *Corresponding Author e-mail: gaganjot@mru.edu.in*

One of the major common assaults in the current Internet of things (IoT) network-based healthcare infrastructures is distributed denial of service (DDoS). The most challenging task in the current environment is to manage the creation of vast multimedia data from the IoT devices, which is difficult to be handled solely through the cloud. As the software defined networking (SDN) is still in its early stages, sampling-oriented measurement techniques used today in the IoT network produce low accuracy, increased memory usage, low attack detection, higher processing and network overheads. The aim of this research is to improve attack detection accuracy by using the DPTCM-KNN approach. The DPTCM-KNN technique outperforms support vector machine (SVM), yet it still has to be improved. For healthcare systems, this work develops a unique approach for detecting DDoS assaults on SDN using DPTCM-KNN.

Keywords: software-defined networking, *k*-nearest neighbors, distributed denial of service, DPTCM-KNN approach, SVM.

1. Introduction

For a long time, DDoS attacks have been a major challenge for Internet users and researchers. Recently, DDoS attacks have increased drastically. When resources or bandwidth of a targeted system are flooded by attack traffic, this is known as a DDoS attack. Although conventional DDoS attacks have well-described traffic patterns, it is difficult to fight against them in real time. Rather than directly flooding victims, these types of attacks flood Internet service provider (ISP) backbone lines, resulting in a large number of attack flows across the victims' Internet connections. The isolation of victim networks from the Internet occurs due to the congestion of the connections. Distinct bots are used by the attackers to create low-rate traffic having actual IP addresses, making identification difficult. These attacks have attracted researchers' attention, and motivated several plausible approaches to deal with them [17–19]. However, the authors of this article are not aware of any traditional deployment that can successfully protect against these DDoS attacks at this time. Anomaly and signature-oriented methods categorize DDoS attack detection methods [6]. This enables the distinction between legitimate and malicious (or aberrant) traffic. The conventional DDoS protection techniques [1, 3] have a variety of drawbacks. To begin with, they frequently need expensive hardware appliances, resulting in additional implementation costs and sophisticated routing hacks [14, 15, 26]. Such attacks are assumed to be stealthy, and the attack traffic might resemble harmless traffic characteristics, and therefore enforcing real-time security against identified assaults is exceedingly challenging [14, 17].

The use of SDN switches to gather and provide essential flow information to the SDN controller is a natural design choice. A software-defined networking administration relies on the network controller. A role of a highly-scalable server is to manage, configure, monitor, and troubleshoot digital network infrastructure from a central, programmable point of control. An SDN-oriented strategy can be theoretically useful in detecting and defending against DDoS attacks by leveraging the network-wide flow characteristics, and such approaches have already been presented in [12, 15, 17, 27]. Although the implementation specifics for detecting attacks vary, they all follow an identical design philosophy and architecture. One of the most important aspects of the design is determining which flow data is required and should be given to the controller. Even though these deployments have offered a variety of benefits, they face certain basic difficulties [20]. First, an SDN switch can simply report flow counter information and is not meant to do complex activities such as flow information pre-processing [28]. As a result, one must rely on other components (such as appliances) to finish them with the help of unmodified commercial off-the-shelf (COTS) SDN switches, which adds to implementation costs. Second, switch flow pre-processing may lose essential original information, causing the controller to exclude attack flows by mistake. This is extremely dangerous when non-link-flooding attacks take place. As a result, such techniques may not be capable of detecting specific assaults, and the accuracy of detection is uncertain.

By examining how DDoS assaults operate, DDoS attack techniques in an SDN environment vary from those in existing networks [22]. DDoS detection and protection technologies are established in a conventional network context [9]. In SDN, the network intrusion detection technique based on the machine learning (ML) may also be used to identify DDoS attacks. ML algorithms can categorize traffic based on flow characteristics and automatically create classification methods based on training data. Approach processes switch information using a NOX controller and handle traffic analysis using a self-organizing map (SOM).

SOM represents a lightweight DDoS detection artificial neural network (ANN) that is unsupervised and competitive learning. Furthermore, the k-nearest neighbor method (KNN) describes an efficient and simple ML technique for classifying flows [23]. Another abnormal traffic detection method called DPTCM-KNN can increase the accuracy of abnormal flow identification while lowering the false alarm rate in the DDoS detection process and the SDN controller's workload while also improving detection accuracy and efficiency. ML algorithms can classify traffic based on flow parameters and automatically generate classification techniques based on training data. Such an approach uses a NOX controller to handle switch knowledge and an AutoMap to assess SOM. This allows security systems to examine and learn from trends in order to detect and prevent similar assaults and adapt to changing behaviour.

The healthcare business is beset by many cyber security vulnerabilities [24]. These challenges vary from ransomware that impugns system stability and patient confidentiality to DDoS assaults that impair institutions' capacity to deliver healthcare services. Fog computing is a new trend in the healthcare industry for emergency patient care. By enhancing the quality of services in the software defined network, fog computing improves healthcare outcomes [13].

While similar assaults occur in other relevant sectors, the architecture of the healthcare system presents distinct obstacles. Cyber-attacks in the medical field can have far-reaching consequences outside monetary loss and data leaks. For example, ransomware is an especially dangerous type of malware since the loss of medical data can endanger lives [26]. This research article is organized as follows. Section 1 discusses the model along with various DDOS detection techniques, and Sec. 2 describes the related works. The methodology is described in Sec. 3. Section 4 presents a DPTCM-KNN-based DDoS attack detection. Section 5 summarizes the outcomes and research findings. Section 6 concludes with some remarks, and Sec. 7 describes future research.

2. Related works

In 2020, Pérez-Díaz *et al.* [1] demonstrated a flexible modular architecture for detecting and mitigating LR-DDoS attacks in SDN environments. The intrusion detection system IDS was trained with the help of six ML models (J48, random tree, REP tree, random forest, multi-layer perceptron (MLP), and support vector machines (SVM)), and their performance was assessed by using the Canadian Institute of Cybersecurity (CIC) DoS dataset. Apart from the difficulties of identifying LR-DoS attacks, the results of the study showed that this technique had a detection rate of 95%. The open network operating system (ONOS) controller was also utilized in the deployment. This proved the architecture's usefulness in detecting and mitigating LR-DDoS attacks.

In 2018, Zheng *et al.* [2] suggested reinforcing anti-DDoS actions in real-time (RADAR) to detect and control DDoS attacks. It was a realistic solution that could fight against a variety of flooding-oriented DDoS attacks, such as link flooding (including Crossfire), SYN flooding, and UDP-based amplification attacks. It effectively identified attacks by detecting attack characteristics in unusual flows and locating attackers using an adaptive correlation analysis to restrict attack traffic. This technique was shown to be capable of detecting and efficiently defending against a variety of DDoS attacks while having a reasonable overhead.

In 2020, Tan *et al.* [3] presented a system for detecting and defending DDoS attacks in the SDN context. The controller would take appropriate defensive steps in response to the attacks. A novel framework of data plane detection and cooperative control plane approaches was developed, which substantially enhanced detection efficiency and accuracy while also preventing DDoS attacks on SDN.

In 2020, Ujjan *et al.* [4] presented adaptive polling-oriented and sFlow sampling with Snort IDS and deep learning-oriented method, which helped to reduce various types of DDoS attacks within the IoT network. SDN's flexible decoupling property enabled network devices to be programmed for needed parameters without the use of proprietary hardware or software from third parties. In control-plane, Snort IDS was used together with the stacked autoencoders (SAE) deep learning method to improve detection accuracy. Moreover, after applying performance measurements in obtained traffic streams, the trade-off between resource overhead and attack detection accuracy was quantitatively examined. When compared to adaptive polling, the suggested system exhibited greater detection accuracy in the sFlow-oriented implementation.

In 2019, Bawany and Shamsi [5] described SEAL (SEcure and AgiLe) – a unique SDN-oriented adaptive architecture for defending smart city applications from DDoS attacks. To improve the security, the SEAL architecture used important SDN properties such as "global visibility, centralized control, and programmability". Furthermore, the SEAL framework's naturally distributed design assured the smart city's "fault tolerance, scalability, and reliability". "D-Defense, A-Defense, and C-Defense" were the three modules that made up the SEAL structure. SEAL's adaptability was accomplished using a modified form of "estimated-weighted moving average (EWMA) filters". To calculate the dynamic threshold in real time, three forms of filters, "proactive filter, active filter, and passive filter", were suggested and developed. An experimental assessment was undertaken. The SEAL framework was designed to secure smart city applications, although it may be used in a variety of other systems.

In 2020, Harikrishna and Amuthan [6] suggested a "convolution recursively enhanced self-organizing map and software defined networking-based mitigation scheme (CRESOM-SDNMS)" for ensuring a better detection rate of preventing DDoS attacks in clouds. The presented CRESOM-SDNMS provided a predominant alternative in handling the conflict of vector quantization with the superior initialization method and increased topology preservation. The simulation tests and findings of the suggested CRESOM-SDNMS revealed a higher classification accuracy.

In 2018, Bhushan and Gupta [7] discussed some key aspects of SDN that make it a good networking solution for cloud computing. A mathematical model based on queuing theory was also used to depict the flow table space of a switch. In addition, a unique flow-table sharing technique was developed to safeguard the SDN-oriented cloud against DDoS attacks caused by flow table overloading. To protect the switch's flow-table from overloading, this technique used the idle flow-tables of various OpenFlow switches present in the network. With minimum participation from the SDN controller, this method enhanced the cloud system's resistance to DDoS assaults. As a result, the communication overhead was relatively low. The claims were backed up by the extensive simulation-oriented tests.

In 2018, Kalkan *et al.* [8] described and tested a joint entropy-based security strategy (JESS) to improve SDN security in order to build a stronger SDN architecture that can withstand DDoS attacks. The suggested approach, in specific, offered a statistical strategy to detect and minimize these risks. As it was based on a statistical model, it was able to neutralize both known and unknown threats.

2.1. Problem definition

Despite significant research, DDoS attacks remain a challenging subject [21]. Existing methods are incapable of identifying DDoS attacks. In specific, the novel sophisticated DDoS attacks based on "low-rate and short-lived" "benign" traffic are difficult to detect. For healthcare systems, this research develops a unique approach for detecting DDoS assaults on SDN using DPTCM-KNN. The proposed DPTCM-KNN classifier efficiency to define its improvement is assessed in the final phase by comparing it to current techniques presented in numerous existing studies. Existing techniques have less efficiency with respect to the proposed technique. As the attack traffic might be masked as harmless traffic, enforcing real-time protection to restrict these discovered assaults is challenging [25]. SDN provides a new way to handle these problems. Table 1 shows the features and challenges of traditional DDoS attack detection on SDN.

ML [1] reduces the entire identified attacks, and the outcomes are migrated to the real-world production environment. However, newer deep learning and ML approaches are not involved. Adaptive correlation analysis [2] does not re-

Table 1. Features and challenges of traditional DDoS attack detection in SDN.

Author	Methodology	Features	Challenges
Pérez-Díaz *et al.* [1]	ML	• The outcomes are migrated to the real-world production environment. • The entire identified attacks are reduced.	• It does not involve newer deep learning and ML approaches.
Zheng *et al.* [2]	Adaptive correlation analysis	• It effectively detects several attacks within fewer delays. • It does not require extra appliances for detecting attacks.	• The effort of the controller rises with larger-scale network traffic.
Tan *et al.* [3]	K-means and KNN	• The controller resources are saved by the detection trigger mechanism. • It uses the respective attack defense measures.	• The burden of a single controller is not minimized by the streaming computing technology.
Ujjan *et al.* [4]	SAE	• It is better with respect to network overhead, low CPU, and accuracy. • It offers efficient and flexible data handling for the DDoS classification.	• The real-time traffic streams are not implemented to reduce the crucial overhead.
Bawany and Shamsi [5]	EWMA	• It is more customizable for attaining the security needs of several applications. • The framework is modeled in modular form.	• Legitimate traffic is not protected against the malicious traffic.
Harikrishna and Amuthan [6]	CRESOM-SDNMS	• The local minimum is minimized in the quantization error. • The data traffic flows are accurately investigated.	• It does not combine SDN and neighborhood function for better malicious data traffic flow detection.
Bhushan and Gupta [7]	Cloud computing	• The resistance of the SDN is enhanced against the attacks. • The used as well as the unused flow table space is approximated by a mathematical model.	• It does not consider the unsupervised deep learning models for the mitigation process.
Kalkan *et al.* [8]	JESS	• It can function effectively with less processing needs and storage. • It returns a better success rate for an unfamiliar generic attack.	• The extensive scale-up for the very large-scale networks is not considered.

quire extra appliances to detect the attacks and effectively detects several attacks within fewer delays. Still, the effort of the controller rises with larger-scale network traffic. K-means and KNN [3] use the respective attack defense measures, and the controller resources are saved by the detection trigger mechanism. Yet, the burden of a single controller is not minimized by the streaming computing technology. SAE [4] offers efficient and flexible data handling for the DDoS classification and is better with respect to network overhead, low CPU, and accuracy. But, the real-time traffic streams are not implemented for reducing the crucial overhead. EWMA [5] models the framework in modular form and is more customizable for attaining the security needs of several applications. Still, the legitimate traffic is not protected against the malicious traffic. CRESOM-SDNMS [6] accurately investigates the data traffic flows and also minimizes a local minimum in the quantization error. Yet, SDN and neighborhood function are not combined for better malicious data traffic flow detection. Cloud computing [7, 29] approximates a mathematical model's used and unused flow table space and enhances the resistance of the SDN against attacks. However, the unsupervised deep learning models are not considered for the mitigation process. JESS [8] returns a better success rate for an unfamiliar generic attack and can function effectively with less processing needs and storage. Still, it does not consider the extensive scale-up for the very large-scale networks. Security protocols may be conceived as a collection of principals that communicate with one another. The protocol's security aims are hoped to be met by forcing agents to provide a chain of structured and encrypted messages [10]. Machine learning allows security systems to examine and learn from trends in order to detect and prevent similar assaults and adapt to changing behavior. Therefore, it is necessary to develop new deep learning approaches for mitigating and detecting DDoS attacks in SDN in an efficient way.

3. Methods and methodology

3.1. Proposed architecture

The primary data is transmitted at a very fast speed through the OpenFlow switch in the SDN architecture. The SDN handles substantial network traffic by locating entries in the flow table, where the packet is sent to many interfaces by the flow entry. The actions, counts, and header field form each entry. Every flow table has several flow entries. The entries in the flow table give the rules for forwarding the data. Figure 1 depicts the suggested DDoS detection architecture using SDN.

Collecting flow states, characteristic value extraction, and classifier judgment constitute the attack detection flow. The collection of flow state delivers a flow

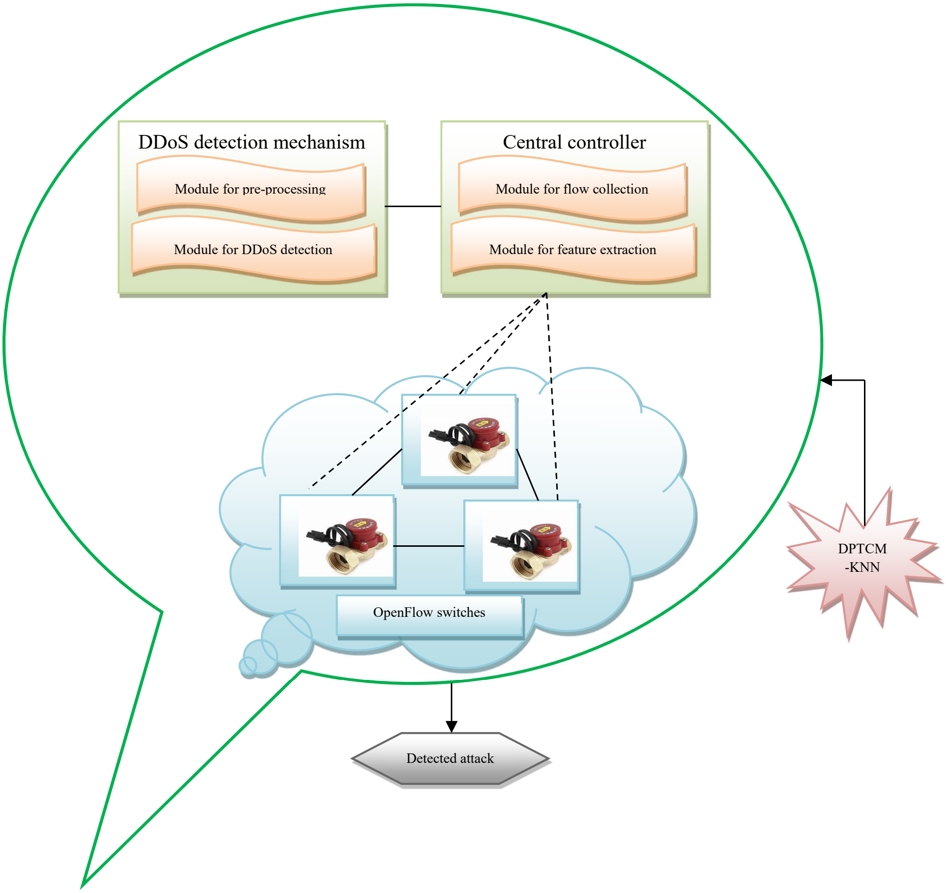
DDoS detection mechanism
Module for pre-processing
Module for DDoS detection
Central controller
Module for flow collection
Module for feature extraction
OpenFlow switches
DPTCM
-KNN
Detected attack

FIG. 1. Introduced architectural model.

table request to the OpenFow switch, and the switch replies to the collection of flow state. Thanks to the SDN architecture, an SDN controller can handle a wide range of data plane resources. There are various data planes, and SDN has the ability to unify and optimize the setup of these disparate resources. The six-tuple characteristic values matrix is used to extract the characteristic values from the characteristic values extraction related to the DDoS attack [11]. The characteristic values information is classified using a DPTCM-KNN-based technique to distinguish between attacking abnormal traffic and normal traffic.

3.2. Dataset description

The dataset used to identify DDoS attacks on SDN is derived from two standard benchmark datasets: the NSL KDD 2000 and the BSUET 2020.

3.2.1. NSL KDD 2000. This dataset is comprised of internet traffic records that have been examined using a simple ID framework. It is referred to as "ghost traffic" and is encountered by traces and real IDS.

3.2.2. BSUET 2020. The data collection is composed of both normal and captured attack traffic. It has six classifications and 1081633 records, 1001984 of which are classified DDoS attacks.

3.3. Extraction of characteristic values from the dataset

The network generates a large number of source IP addresses at random in order to transmit a specified packet size [16]. While detecting the attack, the fluctuation of the source port speed is not always characterized. When the traffic characteristic values are retrieved, the attack procedure generates a large number of new port addresses at random. The DDoS attack strategies in an SDN context differ from those used in traditional networks. DDoS control and prevention technologies are well-established in a traditional network context. In SDN systems, DDoS attack detection and protection mechanisms are often built by repurposing techniques from existing networks. Several conventional SDN investigations are studied and differentiated, and data processing and analysis are carried out using information extracted from the flow status based on the previous studies. For identifying the DDoS attack, the six tuple characteristic values listed below were obtained.

3.3.1. SSIP. It is described as the count of source IP addresses per time unit, as shown in Eq. (1):

$$\mathrm{SSIP} = \frac{Sm\_\mathrm{IP}_{srcb}}{TB}, \tag{1}$$

where the source IP number is Sm_IP_{srcb} and the sampling interval is TB. When an attack happens, an attacker generates a large number of attacks in order to transmit data packets in a random order, and the source IP address count rapidly increases.

3.3.2. SSP. It is described as the count of source ports per time unit, as shown in Eq. (2):

$$\mathrm{SSP} = \frac{Sm\_\mathrm{port}_{srcb}}{TB}, \tag{2}$$

where the attack source port count is Sm_port_{srcb}. When a large number of attack requests are received, a large number of port counts are created at random.

3.3.3. SDFP. It is described as the standard deviation of the number of packets in the *TA* period, as shown in Eq. (3):

$$\mathrm{SDFP} = \sqrt{\frac{1}{NB}\sum_{ib=1}^{NB}(\mathrm{packets}_{ib} - \mathrm{Mean\_packets})^2}, \tag{3}$$

where the period is *TB* and the average packet count is

$$\mathrm{Mean\_packets} = \left(\frac{1}{NB}\right)\sum_{ib=1}^{NB}\mathrm{packets}_{ib}.$$

When an attack happens, the overall flow entry count per period is determined by *NB*. For generating the attack effect, the general attack data packets are very tiny, and the SD associated with the flow packets is smaller than the regular flow.

3.3.4. SDFB. It is described as the count of bits in the *TA* period, as shown in Eq. (4):

$$\mathrm{SDFB} = \sqrt{\frac{1}{NB}\sum_{ib=1}^{NB}(\mathrm{bytes}_{ib} - \mathrm{Mean\_bytes})^2}, \tag{4}$$

where the average o6f the bit count is

$$\mathrm{Mean\_bytes} = \left(\frac{1}{NB}\right)\sum_{ib=1}^{NB}\mathrm{bytes}_{ib}.$$

When a certain event happens, the packet load is reduced by delivering fewer data packets.

3.3.5. SFE. It is described as the count of flow entries per time of unit, as shown in Eq. (5):

$$\mathrm{SFE} = \frac{NB}{TB}. \tag{5}$$

3.3.6. RPF: It is described as the ratio of interactive to total flow entries, as shown in Eq. (6):

$$\mathrm{RPF} = \frac{2 * \mathrm{Pair\_}Sm}{NB}, \tag{6}$$

where the interactive flow entry count is Pair_*Sm*. The source host passes a request to the destination site that includes the requirements listed below.

The source IP of *pkt_ib* is similar to the destination IP of *pkt_jb*. The destination port count of *pkt_ib* is identical to the source port count of *pkt_jb*. The destination IP of *pkt_ib* is identical to the source IP of *pkt_jb* and the source port count of *pkt_ib* is identical to the destination port count of *pkt_jb*". There exist two interactive flow entries, and they fulfil Eq. (7):

$$\begin{aligned} Sce_{\mathrm{IP}_{ib}} &= Des_{\mathrm{IP}_{jb}}, \\ Sce_{\mathrm{port}_{ib}} &= Des_{\mathrm{port}_{jb}}, \\ Sce_{\mathrm{IP}_{jb}} &= Des_{\mathrm{IP}_{ib}}, \\ Des_{\mathrm{port}_{jb}} &= Des_{\mathrm{IP}_{ib}}. \end{aligned} \tag{7}$$

When an attack occurs, the flow entries increase dramatically. The destination host is unable to reply to the interactive flow in a timely manner. Normally, the attacker employs large fake source addresses during the attack procedure.

4. DPTCM-KNN-based DDoS attack detection

This work develops architecture for identifying anomalous flows in an SDN environment and presents the DPTCM-KNN anomaly flow detection algorithm to address the shortcomings of SDN-based flow detection techniques. It uses the notion of transductive confidence machines (TCM) to obtain the level of confidence of a detecting point, and it uses the probability to decide if the detecting point is subjected to the group, according to stochastic algorithm theory. The higher the p value, the more probable the detection point corresponds to the category. The technique increases the accuracy of anomalous flow detection by using strangeness and independence as its dual inspection standards, which are the TCM-KNN algorithm's detection loopholes.

The primary drawback of KNN-oriented algorithms is that they are lazy learners, meaning that they do not learn anything from the training data and hardly classify the training data. The KNN method will locate the k-nearest neighbors to the new instance from the training data and set the anticipated class label as the most frequent label among the k-nearest surrounding points to forecast the label of a new instance. The primary downside of this technique is that for each prediction, the algorithm must compute the distance and sort all of the training data, which may be time-consuming if there are many training instances. Another downside of this technique is that the algorithm does not learn anything from the training data, which can lead to problems with generalization and robustness to noisy data. Furthermore, altering k affects the projected class label. It makes sense to use eager learning to solve the fundamental challenge of anomaly identification in SDN networks because patterns of abnormal behavior change rapidly.

As a result, we consider integrating the DPTCM (strangeness and independence as an inspection standard) with eager learning algorithms such ANN, SVM, and random forest. The main advantage of employing an eager learning approach, such as ANN, is that the objective function is estimated globally during the process of training, which saves a lot of space compared to a lazy learning system. Eager learning algorithms are also much better at dealing with noise in the training data. Eager learning is an instance of offline learning, where post-training queries to the system do not affect the system, and thus the same query to the system always produces the same result. Figure 2 depicts the process of detecting DDoS attacks using a DPTCM-KNN classifier.

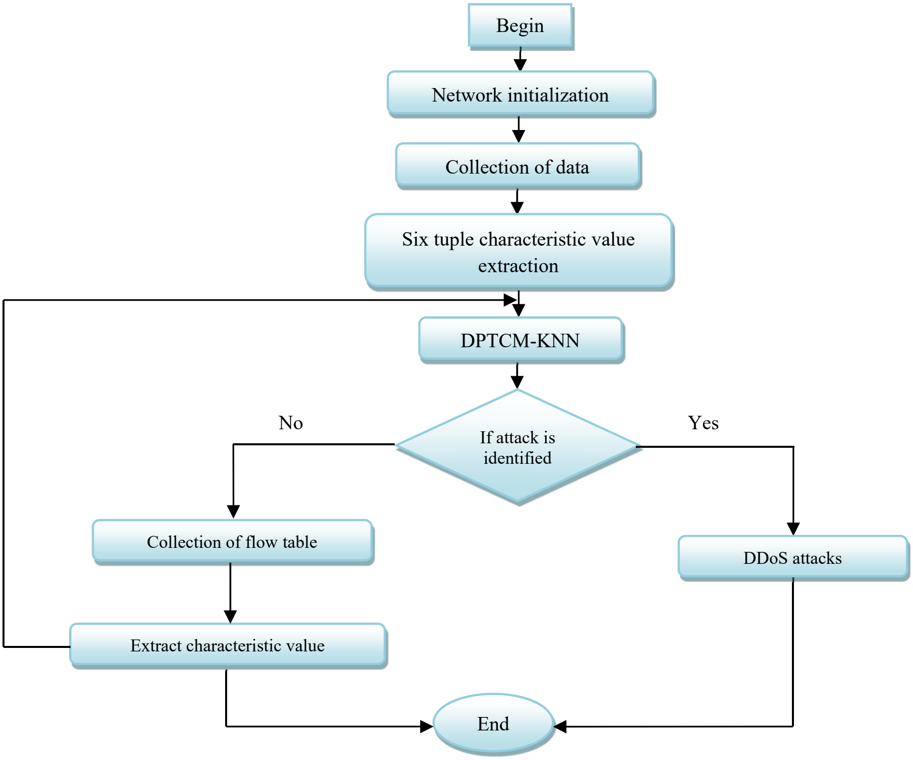

Fig. 2. A DDoS attack detection in SDN by the DPTCM-KNN classifier.

5. Results and discussions

The suggested model was implemented using the DPTCM-KNN in MATLAB 2020a, and the results were obtained. Consequently, the DPTCM-performance

KNN's analysis was clearly better in detecting DDoS assaults on SDN. This study also accomplishes outstanding DDoS attack detection because of the efficient features of DPTCM-KNN. MathWorks' MATLAB represents a numeric computing environment and proprietary multi-paradigm programming language. It allows for the design of user interfaces, algorithm development, data and function graphing, matrix operations, and interfaces with programs written in a variety of languages. The performance was conducted in terms of TPR for the BUET dataset. Table 2 reveals the simulation parameters that were used to run the suggested attack detection model.

Table 2. Simulation parameters.

Parameters	Values
Number of primary users	5
Number of secondary users	10
Number of malicious users	4
Number of iterations	100
Number of samples	4000

5.1. Performance metrics

The different measures used here are as follows.

5.1.1. Accuracy. It is defined as the discrepancy in the recognized outcome to the ground value:

$$\mathrm{Acry} = \frac{\mathrm{TPT} + \mathrm{TNT}}{\mathrm{TPT} + \mathrm{TNT} + \mathrm{FPT} + \mathrm{FNT}}, \tag{8}$$

where the terms TPT, TNT, FPT, and FNT represent the true positive, true negative, false positive, and false negative, respectively.

5.1.2. TPR. It is described as the probability that an actual positive will test positive:

$$\mathrm{TPR} = \frac{\mathrm{TPT}}{\mathrm{TPT} + \mathrm{FNT}}, \tag{9}$$

where the terms TPT and FNT represent the true positive and false negative, respectively.

5.1.3. FPR. It refers to the total positive results within the negative output:

$$\mathrm{FPR} = \frac{\mathrm{FPT}}{\mathrm{FPT} + \mathrm{TNT}}. \tag{10}$$

5.2. Analysis of TPR

The TPR analysis of the BEUT and NSL KDD dataset in the case of 4000 samples for the proposed and existing methods is shown in Fig. 3a. For 1000 samples, in the case of the BUET dataset, the TPR of DPTCM-KNN is 15.85%, 11.76%, 6.74% and 4.40% higher than ABTSVM, KNN-ACO, TCM-KNN, and DPTCM-SVM, respectively. When considering 2000 samples, the TPR of DPTCM-KNN is improved by 11.76%, 9.20%, 6.74%, and 4.40% for ABTSVM, KNN-ACO, TCM-KNN, and DPTCM-SVM, respectively. In the case of 3000 samples, the TPR of DPTCM-KNN is 11.36%, 7.69%, 5.38%, and 4.26% better than ABTSVM, KNN-ACO, TCM-KNN, and DPTCM-SVM, respectively. While considering the NSL KDD dataset as shown in Fig. 3b, for 4000 samples, the TPR of the DPTCM-KNN is 1.55%, 3.14%, 5.91%, and 8.84% more advanced than DPTCM-SVM, TCM-KNN, KNN-ACO, and ABTSVM, respectively. Thus, it is clear that the TPR analysis provides better outcomes using the DPTCM-KNN method than the traditional approaches when it is estimated with TPR for both datasets.

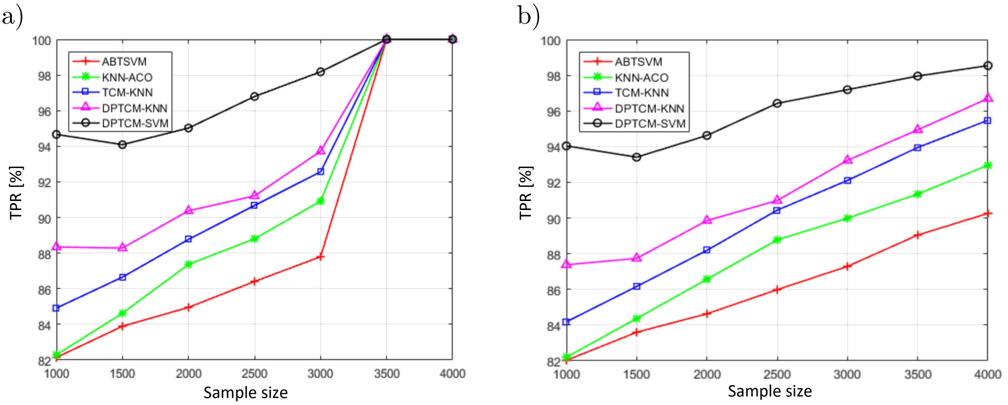

Fig. 3. The TPR analysis of the proposed and existing DDoS attack detection on SDN for a) BEUT dataset and b) NSL KDD dataset.

5.3. Analysis of FPR

The FPR analysis of the proposed and existing methods for the DDoS attacks on SDN considering 4000 samples on two datasets is given in Fig. 4. In Fig. 4a, for the BEUT dataset, the FPR of the DPTCM-KNN for 3000 samples is 4.26%, 5.95%, 7.69%, and 12% more advanced than DPTCM-SVM, TCM-KNN, KNN-ACO, and ABTSVM, respectively. While considering Fig. 4b, for the NSL KDD dataset, the FPR of DPTCM-KNN for the 4000th sample is 46.15%, 70.83%, 68.18%, and 66.67% higher than DPTCM-SVM, TCM-KNN, KNN-ACO, and ABTSVM, respectively. Thus, the FPR analysis produces good outcomes with

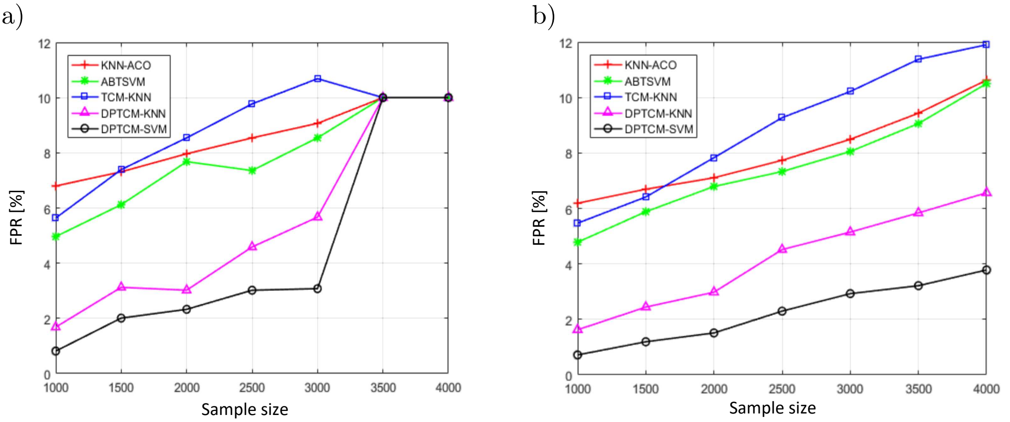

Fig. 4. The FPR analysis of the proposed and existing DDoS attack detection on SDN for a) BEUT dataset and b) NSL KDD dataset.

the proposed method compared to other techniques for the DDoS attacks on SDN for both datasets.

5.4. Analysis of accuracy

The accuracy analysis for the DDoS attacks on SDN for the proposed and conventional techniques for both datasets is given in Fig. 5. In Fig. 5a, for the BEUT dataset, the accuracy of the DPTCM-KNN at the 3000th sample is 5.26%, 7.53%, 8.11%, and 12.36% better than DPTCM-SVM, TCM-KNN, KNN-ACO, and ABTSVM respectively. When considering Fig. 5b, for the NSL KDD dataset, the accuracy of DPTCM-KNN at 4000th sample is 1.52%, 4.17%, 5.82%, and 8.11% better than DPTCM-SVM, TCM-KNN, KNN-ACO, and ABTSVM, re-

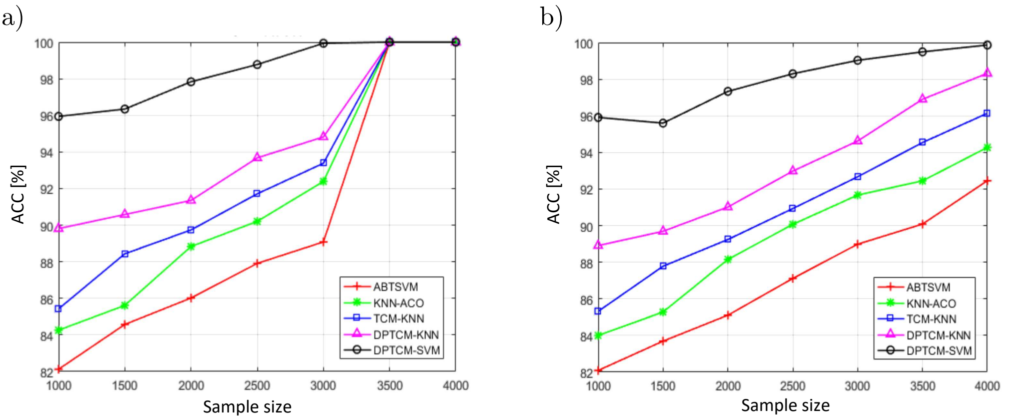

Fig. 5. Accuracy analysis of the proposed and existing DDoS attack detection on SDN for a) BEUT dataset, and b) NSL KDD dataset.

spectively. Hence, the outcomes of the accuracy analysis are better with the DPTCM-KNN than other methods for the DDoS attack detection on SDN in both datasets. This model is more accurate in terms of working efficiency, although using the DPTCM-KNN enhanced attack detection accuracy, the DPTCM-KNN approach has a better results than SVM, which has to be improved.

5.5. Classifier analysis

The classifier analysis in terms of accuracy for different classifier algorithms is depicted in Fig. 6. Here, the DPTCM-KN and SVM achieved a high accuracy of 0.98% followed by ABTSVM, KNN-ACO, and TCM-KNN with 0.95%, 0.93%, and 0.92%, respectively. Thus, it is clear that the classifier analysis produces better outcomes than the other methods in terms of accuracy for the DDoS attack detection on SDN.

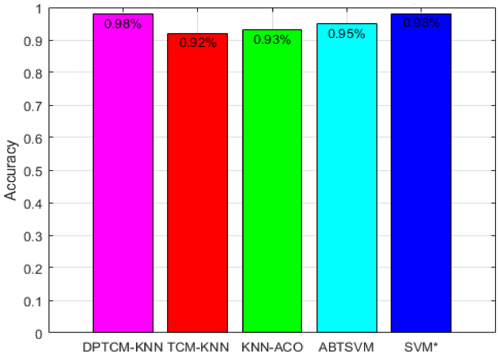

Fig. 6. Classifier analysis of the proposed and existing DDoS attack detection on SDN.
* SVM was run on ~100k instances, others only on 4000.

6. Conclusions

This research produced a novel method for detecting DDoS attacks on SDN using DPTCM-KNN for healthcare systems. In the last phase, the efficiency of the proposed DPTCM-KNN classifier was determined by comparing it to existing approaches in terms of several analyses for defining its improvement. From the analysis, for 1000 samples, in the case of the BUET dataset, the TPR of DPTCM-KNN was 15.85%, 11.76%, 6.74% and 4.40% higher than ABTSVM, KNN-ACO, TCM-KNN, and DPTCM-SVM, respectively. The complexity of this method is that it requires the algorithm to compute the distance and sort all of the training data for each prediction, which might take a long time if there are a lot of them. As a result, it was apparent that the DPTCM-performance KNN's analysis was superior in terms of identifying DDoS attacks on SDN.

Due to the efficient features of DPTCM-KNN, this study also achieved excellent DDoS attack detection. It had a high level of attack detection accuracy and did not require training datasets containing malicious data. In the future, new types of attacks can be identified to further improve the IoT-based health sector. Although this study produced an increased attack detection accuracy by using the DPTCM-KNN method, and the DPTCM-KNN technique has a superior performance compared to SVM, it still needs to be improved.

7. Future research

Future research on preventing DDoS attacks on SDN might go in one of those directions. Anomaly detection methods for SDN currently available have limited accuracy and are less real-time effective. In addition, they do not facilitate gradual learning. As a result, improving and optimizing the conventional method and establishing real-time identification and high precision models for anomalous flows are critical for SDN-oriented flow detection to adapt to large-scale networks. While individual client flows may be insignificant, traffic abnormalities using aggregated traffic may be catched. As the identified abnormalities do not alert the controller, it is unable to detect attack activity. The most frequent benefits of SDN include healthcare programmability, agility, the ability to build policy-driven network monitoring, and the ability to apply methods to control. Its most significant advantage is that it enables the development of a framework to accommodate more data-intensive applications such as big data and virtualization. Cluster analysis is an unsupervised learning technique that does not need any training, yet there is no discernible difference in the probability density distribution within aberrant and normal data, particularly hidden worms and sluggish DDoS. When abnormal points serve as a dividing line within abnormal and normal points, a decision must be made based on the relative deviation. The algorithm's precision must be enhanced further since it contains flaws in anomaly detection of the detection sites. The controllers are put under a lot of stress by the SDN-oriented flow detection approach. The future scope of this research will focus on new types of assaults that may be discovered in the future, allowing the IoT-based health sector to improve even more. Although using the DPTCM-KNN approach enhanced attack detection results, and the DPTCM-KNN approach outperforms SVM, it still has to be improved. With more network flows, controllers must monitor the whole network environment, and identify flow abnormalities in the controllers' workload. They are also limited in scalability, accuracy, and efficiency when it comes to detecting huge and fast data flows As a result, more precise detection architectures for SDN anomalies are required, and therefore, in recent years, SDN-oriented anomalous flow detection has become a popular topic in recent years.

References

1. J.A. Pérez-Díaz, I.A. Valdovinos, K.-K.R. Choo, D. Zhu, A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning, *IEEE Access*, **8**: 155859–155872, 2020, doi: 10.1109/ACCESS.2020.3019330.

2. J. Zheng, Q. Li, G. Gu, J. Cao, D.K.Y. Yau, J. Wu, Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis, *IEEE Transactions on Information Forensics and Security*, **13**(7): 1838–1853, 2018, doi: 10.1109/TIFS.2018.2805600.

3. L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, Y. Deng, A new framework for DDoS attack detection and defense in SDN environment, *IEEE Access*, **8**: 161908–161919, 2020, doi: 10.1109/ACCESS.2020.3021435.

4. R.M.A. Ujjan, Z. Pervez, K. Dahal, A.K. Bashir, R. Mumtaz, J. González, Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN, *Future Generation Computer Systems*, **111**: 763–779, 2020, doi: 10.1016/j.future.2019.10.015.

5. N.Z. Bawany, J.A. Shamsi, SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks, *Journal of Network and Computer Applications*, **145**: 102381, 2019, doi: 10.1016/j.jnca.2019.06.001.

6. P. Harikrishna, A. Amuthan, SDN-based DDoS attack mitigation scheme using convolution recursively enhanced self organizing maps, *Sādhanā*, **45**: Article No. 104, 2020, doi: 10.1007/s12046-020-01353-x.

7. K. Bhushan, B.B. Gupta, Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment, *Journal of Ambient Intelligence and Humanized Computing*, **10**: 1985–1997, 2019, doi: 10.1007/s12652-018-0800-9.

8. K. Kalkan, L. Altay, G. Gür, F. Alagöz, JESS: Joint entropy-based DDoS defense scheme in SDN, *IEEE Journal on Selected Areas in Communications*, **36**(10): 2358–2372, 2018, doi: 10.1109/JSAC.2018.2869997.

9. N. Agrawal, S. Tapaswi, Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges, *IEEE Communications Surveys & Tutorials*, **21**(4): 3769–3795, 2019, doi: 10.1109/COMST.2019.2934468.

10. Y. Xiang, K. Li, W. Zhou, Low-rate DDoS attacks detection and traceback by using new information metrics, *IEEE Transactions on Information Forensics and Security*, **6**(2): 426–437, 2011, doi: 10.1109/TIFS.2011.2107320.

11. C. Zhang, Z. Cai, W. Chen, X. Luo, J. Yin, Flow level detection and filtering of low-rate DDoS, *Computer Networks*, **56**(15): 3417–3431, 2012, doi: 10.1016/j.comnet.2012.07.003.

12. W. Zhijun, X. Qing, W. Jingjie, Y. Meng, L. Liang, Low-rate DDoS attack detection based on factorization machine in software defined network, *IEEE Access*, **8**: 17404–17418, 2020, doi: 10.1109/ACCESS.2020.2967478.

13. A. Kishor, C. Chakraborty, W. Jeberson, Reinforcement learning for medical information processing over heterogeneous networks, *Multimedia Tools and Applications*, **80**(16): 23983–24004, 2021, doi: 10.1007/s11042-021-10840-0.

14. J. Ye, X. Cheng, J. Zhu, L. Feng, L. Song, A DDoS attack detection method based on SVM in software defined network, *Security and Communication Networks*, **2018**: Article ID 9804061, 8 pages, 2018, doi: 10.1155/2018/9804061.

15. N. Hoque, D.K. Bhattacharyya, J.K. Kalita, Botnet in DDoS attacks: Trends and challenges, *IEEE Communications Surveys Tutorials*, **17**(4): 2242–2270, 2015, doi: 10.1109/COMST.2015.2457491.

16. R.M.A. Saad, M. Anbar, S. Manickam, E. Alomari, An intelligent ICMPv6 DDsS flooding-attack detection framework (v6IIDS) using back-propagation neural network, *IETE Technical Review*, **33**(3): 244–255, 2016, doi: 10.1080/02564602.2015.1098576.

17. A. Saied, R.E. Overill, T. Radzik, Detection of known and unknown DDoS attacks using artificial neural networks, *Neurocomputing*, **172**: 385–393, 2016, doi: 10.1016/j.neucom.2015.04.101.

18. T.A. Pascoal, Y.G. Dantas, I.E. Fonseca, V. Nigam, Slow TCAM exhaustion DDoS attack, [in:] S. De Capitani di Vimercati, F. Martinelli [Eds.], *ICT Systems Security and Privacy Protection, SEC 2017. IFIP Advances in Information and Communication Technology*, Vol. 502, pp. 17–31, Springer International Publishing, Cham, 2017.

19. K. Hong, Y. Kim, H. Choi, J. Park, SDN-assisted slow http DDoS attack defense method, *IEEE Communications Letters*, **22**(4): 688–691, 2017, doi: 10.1109/LCOMM.2017.2766636.

20. R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, P. Singh, Prediction of heart disease using a combination of machine learning and deep learning, *Computational Intelligence and Neuroscience*, **2021**: pp. 1–11, A.A. Abd El-Latif [Ed.], 2021, doi: 10.1155/2021/8387680.

21. O.A. Osanaiye, K.-K.R. Choo, M. Dlodlo, Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework, *Journal of Network and Computer Applications*, **67**: 147–165, 2016, doi: 10.1016/j.jnca.2016.01.001.

22. P. Ratta, A. Kaur, S. Sharma, M. Shabaz, G. Dhiman, Application of blockchain and Internet of things in healthcare and medical sector: Applications, challenges, and future perspectives, *Journal of Food Quality*, **2021**: pp. 1–20, 2021, doi: 10.1155/2021/7608296.

23. S. Lim, S. Yang, Y. Kim, S. Yang, H. Kim, Controller scheduling for continued SDN operation under DDoS attacks, *Electronics Letters*, **51**(16): 1259–1261, 2015, doi: 10.1049/el.2015.0334.

24. T. Thakur *et al.*, Gene expression-assisted cancer prediction techniques, *Journal of Healthcare Engineering*, **2021**: Article ID 4242646, 9 pages, D. Zaitsev [Ed.], 2021, doi: 10.1155/2021/4242646.

25. S. Yu, W. Zhou, R. Doss, W. Jia, Traceback of DDoS attacks using entropy variations, *IEEE Transactions on Parallel and Distributed Systems*, **22**(3): 412–425, 2011, doi: 10.1109/TPDS.2010.97.

26. A. Kishor, C. Chakraborty, W. Jeberson, Intelligent healthcare data segregation using fog computing with internet of things and machine learning, *International Journal of Engineering Systems Modelling and Simulation*, **12**(2–3): 188–194, 2021, doi: 10.1504/IJESMS.2021.115533.

27. K. Kalkan, G. Gür, F. Alagöz, Filtering-based defense mechanisms against DDoS attacks: A survey, *IEEE Systems Journal*, **11**(4): 2761–2773, 2017, doi: 10.1109/JSYST.2016.2602848.

28. J. Mirkovic, P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, *SIGCOMM Computer Communication Review*, **34**(2): 39–53, 2004, doi: 10.1145/997150.997156.

29. B. Wang, Y. Zheng, W. Lou, Y.T. Hou, DDoS attack protection in the era of cloud computing and software-defined networking, *Computer Networks*, **81**: 308–319, 2015, doi: 10.1016/j.comnet.2015.02.026.

Received January 2, 2022; revised version January 28, 2022;
accepted February 2, 2022