J. Inst. Eng. India Ser. B (April 2020) 101(2):117–128
https://doi.org/10.1007/s40031-020-00442-z

ORIGINAL CONTRIBUTION

DDoS Detection and Alleviation in IoT using SDN (SDIoT-DDoS-DA)

Azka Wani[1] · **S. Revathi**[2]

Received: 28 July 2018 / Accepted: 11 April 2020 / Published online: 28 April 2020

Abstract The Internet of Things (IoT) is an ever expanding discipline encompassing all orbits of life, and its development has resulted in enormous benefits. IoT has made it possible for simple electronic objects to participate in the Internet. However, the growth of IoT has also resulted in considerable security issues. Devices that build up an IoT network have constrained resources and battery power making it difficult to incorporate a proper security mechanism in an IoT environment. The devices in IoT are vulnerable to numerous threats, and the volume of these threats is ever increasing. Distributed Denial of Service (DDoS) is one of the attacks that have gained momentum with the growth of IoT. DDoS not only influences IoT network, but IoT botnets can also be used to launch voluminous DDoS attacks. Although numerous lightweight security protocols and mechanisms have been designed for improvement of security scenario in IoT networks, most of the security concerns are yet to be assuaged. In this paper, we propose a Software-Defined Network (SDN)-based security mechanism, for detection and alleviation of DDoS in IoT networks. SDN is a flexible method of managing and controlling a network that segregates data and control planes. It makes networks programmable which can be used to develop an efficient method to deal with catastrophic attacks in IoT networks.

✉ Azka Wani
graceazka@gmail.com

[1] Department of Computer Applications, Crescent B S Abdur Rahman Institute of Science and Technology, Vandalur, Chennai 600048, India

[2] Department of Computer Science and Engineering, Crescent B S Abdur Rahman Institute of Science and Technology, Vandalur, Chennai 600048, India

Keywords IoT · Security · SDN · OpenFlow · DDoS · Detection · Alleviation

Introduction

The Internet of Things (IoT) is the network of devices or "Things" that have the ability to transfer information using Internet. The networked objects include sophisticated networking devices as well as devices of day-to-day use that are embedded with tiny sensors [1]. The IoT is expected to encompass billions of devices in near future. The heterogeneity of devices and protocols used in the underlying architecture makes it hard to manage and operate IoT networks [2]. The nodes in an IoT network have limited resources and computational capability. In an IoT scenario, maximum resources are consumed by device functionality and it is difficult to incorporate comprehensive security mechanisms. Add to that the privacy concerns. There is a one-on-one interaction between humans and IoT devices, to the extent that some sensor embedded devices are fitted in vivo as well. These devices expose personal and critical information to the unsupervised world of the Internet. Ensuring security in such a constrained and heterogeneous scenario is a challenging task, yet necessary.

The proliferation of automated devices has resulted in dramatic improvement and profit in almost every sector. The benefits of ongoing miniaturization cannot be ignored [3]. However, the security concerns that if not dealt properly can lead to many incidents of compromise and information theft. Easy accessibility procedures for an IoT environment make it susceptible to numerous security threats, such as Distributed Denial of Service (DDoS), information disclosure, spoofing and elevation of privilege

[4]. The exploitation of IoT infrastructure for launching DDoS attacks has been a major security concern lately. The increasing number of IoT devices is considered as a primary cause for voluminous DDoS attacks of hundreds of Tbps [5]. The IoT devices due to their poor security measures can be attacked with the least efforts. Such devices can also be used to create massive DDoS attacks since the quantity of such devices is increasing exponentially.[6]. France-based hosting provider OVH was the victim of the record-breaking DDoS attacks of 1 Tbps on September 27, 2016, and a DDoS attack of 665Gbps was delivered by a botnet of IoT devices on September 21, 2016 for Krebs on Security Web site. The havoc created by Ransomware in 2017 cannot be ignored until another massive DDoS attack hits the cyber-world. Network security threat has got a new boost with growth and use of IoT [7–9].

The security of IoT is the need of the hour because IoT handles large amounts of sensitive data [10]. The increasing number of voluminous DDoS attacks also necessitates proper security enforcement in IoT. The security measures for traditional networks have evolved over time and provide a relatively comprehensive security mechanisms, but the process of safeguarding IoT is still in the initial stage of development [11]. Many studies have been conducted that address the security concerns in IoT, but little work has been done toward the defense against DDoS attacks in an IoT environment. This paper aims to present a Software-Defined Network (SDN)-based security framework for detection and alleviation of DDoS in IoT architecture (SDIoT-DDoS-DA).

Software-defined networking (SDN) is a novel networking concept that provides flexible network control and management by segregation of data and control planes. The network control and management have been shifted to a centralized control plane called controller, while the switches are limited in their functionality to simple forwarding devices. SDN is gaining popularity and has been implemented in a variety of sectors due of its enhanced network operation and management features. The programmability feature of SDN provides better maintenance of the network, as network administrators are able to control the functioning of the network at application level, instead of configuring each network device separately [12]. SDN is a preferred solution for many network-related challenges in contemporary times. The main goal of SDN is to hide all the complexities of management and control functionality of the system resources from the end users. In this work, we propose an SDN-based security framework for detection and alleviation of DDoS in the IoT networks. This SDN-based security mechanism monitors the traffic from the IoT network and decides whether the network is under a DDoS attack.

The proposed mechanism brings together two innovative technologies—SDN and IoT. Devices in the IoT are limited in computing power and resources. Traditional security methods such as hashing, cryptography or anti-malware cannot be used for such resource-constrained IoT devices. The DDoS attack is one of the powerful attacks which can cause a lot of damage. Even conventional networks require ample effort to mitigate it. Therefore, it is not easy for IoT devices to counter DDoS attacks. However, SDN allows security enforcement for IoT at network infrastructure level. The proposed mechanism does not burden IoT devices with extra processing as it includes security at the gateway.

The contributions of the paper can be summed up as:

- SDN features have been harnessed to mitigate DDoS in IoT networks.
- Micro-Cluster Outlier Detection (MCOD) is used to identify abnormal behavior in IoT networks.
- Multilayer perceptron (MLP), the machine learning approach, decides whether the abnormality has been caused by a DDoS attack.

The rest of the paper is organized as follows: Section two presents IoT device security and related work. Section three focuses on the concept of Software-Defined Networking and its use in problem identification. Section four introduces the proposed security framework. Section five presents the implementation work, performance evaluation, results and discussions. Section six summarizes the study and highlights the areas for further work.

IoT Security and Related Work

This section provides an overview of security-related issues in IoT, current approaches toward security improvisation in IoT. The importance of security within IoT and requirements are given by the end of this section. The devices in an IoT network have varying characteristics and constrained resources; hence, designing a concrete security mechanism asks for a comprehensive precise approach. The heterogeneity of IoT is one such characteristic which results in different processing capabilities of the devices. The communication mediums used by IoT devices have not been standardized yet and function differently [13]. The diversity in communication protocols in IoT makes it difficult to deploy conventional network security systems on the IoT platform.

Security-Related Issues in IoT

The rapid increase in cyber-attacks has been linked to the growth of IoT. The expansion of poorly secured connected objects has resulted in massive catastrophic attacks. The IoT devices are an easy target for launching DDoS attacks, malware infection and botnet creation. The IoT devices carry sensitive information of individuals or patients that can be exploited for privacy attacks and advanced persistent threat (APT) as well. The DDoS attack, in particular, is one of the most threatening attacks that has shaken cyber-world since the advent of IoT [14]. IoT devices are easily overpowered and controlled by hackers for the creation of bogus traffic that eventually forms a DDoS attack. The smart network of IoT has automated every task and carries sensitive information with the least protection. A DDoS attack on such a network can result in an abnormal shutdown of the entire system and can cause collateral damage too. The DDoS attack on an IoT network of a healthcare system can risk the lives of patients, and likewise, such an attack on a vehicular IoT network can cause uncontrolled accidents. The SDN-based IoT simplifies the network management and provides a clear visualization of network resources. Many researchers have suggested methods of protecting IoT networks by utilizing the SDN infrastructure [15]. Some of the recent studies that focus on securing IoT network using SDN and research work carried against DDoS attack in IoT have been summarized below:

SDN- and Non-SDN-Based Security in IoT

Sheikhan et al. [16] introduced a method termed as MOPF for identifying internal and external attacks in an IoT network. Anomaly detection for 6LoWPAN was also proposed by the authors. Salman et al. [17] have used SDN/NFV and cloud/edge computing to create hierarchical security architecture for IoT network. The suggested mechanism consists of six layers (the device layer, the access network layer, the access control layer, the core network layer, the core control layer and the application layer). The architecture is based on the human nervous system and does not have a full centralized control. In this framework, there is one central controller called the core controller which provides global network control and there are access controllers to which the devices are connected. Some other notable researches done in IoT have been presented in [18–21].

DDoS security in IoT

The researches conducted toward DDoS-type attacks in IoT have been presented in [22–25]. Kawamura et al. [22] analyzed DDoS attacks in an IoT network by an event detection module using the data from network time synchronization service. The authors have used Network Time Protocol (NTP). The proposed method is developed for the real-time detection of DDoS in IoT networks.

De Donno et al. [26] have proposed a method called AntibIoTic for securing IoT against DDoS attacks. AntibIoTic searches for poorly secured IoT devices on the Internet. On finding such device, it is compromised and then cleaned to secure its surroundings. At the same time, the owner is made aware of the threat so that some solution is implemented to solve the issue. The device owner uses the proposed guidelines to secure the IoT device and surroundings. Once the device is secured, the device is freed by the AntibIoTic.

Zhangh et al. [27] have introduced a lightweight algorithm for prevention of DDoS attack in an IoT network. The proposed method is deployed on working nodes, which are data collectors, to detect and avoid attacks. The attack detection mechanism which is associated with the working nodes is lightweight.

The Concept of SDN

This section reviews the working of a Software-Defined Network. Separation of data and control planes is basis of the new concept of networking called SDN. There are three planes in SDN architecture. The lowermost plane is the data plane which contains switches that are SDN enabled. The switches are only packet forwarders and have no role to play in decision making. The routing decisions are taken by the control plane which includes controller [28]. The data plane requests the controller to form routing rules. The controller also takes other control-based decisions for the data packets. The third plane in SDN comprises of an application programming interface (API) which contains the applications for controlling the network (Fig. 1).

The controller decides the path of the packets and takes other control decisions according to the application plane. The medium between the data plane and control plane is termed as the southbound interface, while the medium between control plane and application plane is called northbound interface. The communication in the southbound interface is governed by protocols like OpenFlow [29]. It was the first communication protocol used in the southbound interface and has since become a de facto standard. Generally, an OpenFlow-enabled switch contains a flow table that forwards packets as per the flow rules. The flow tables are filled with flow entries. Each flow entry contains *match fields, statistics and actions*. The *match fields* check incoming packets, the *statistics* field keeps count of packets matched by each flow entry and the *actions* field decides the action that has to be taken for each

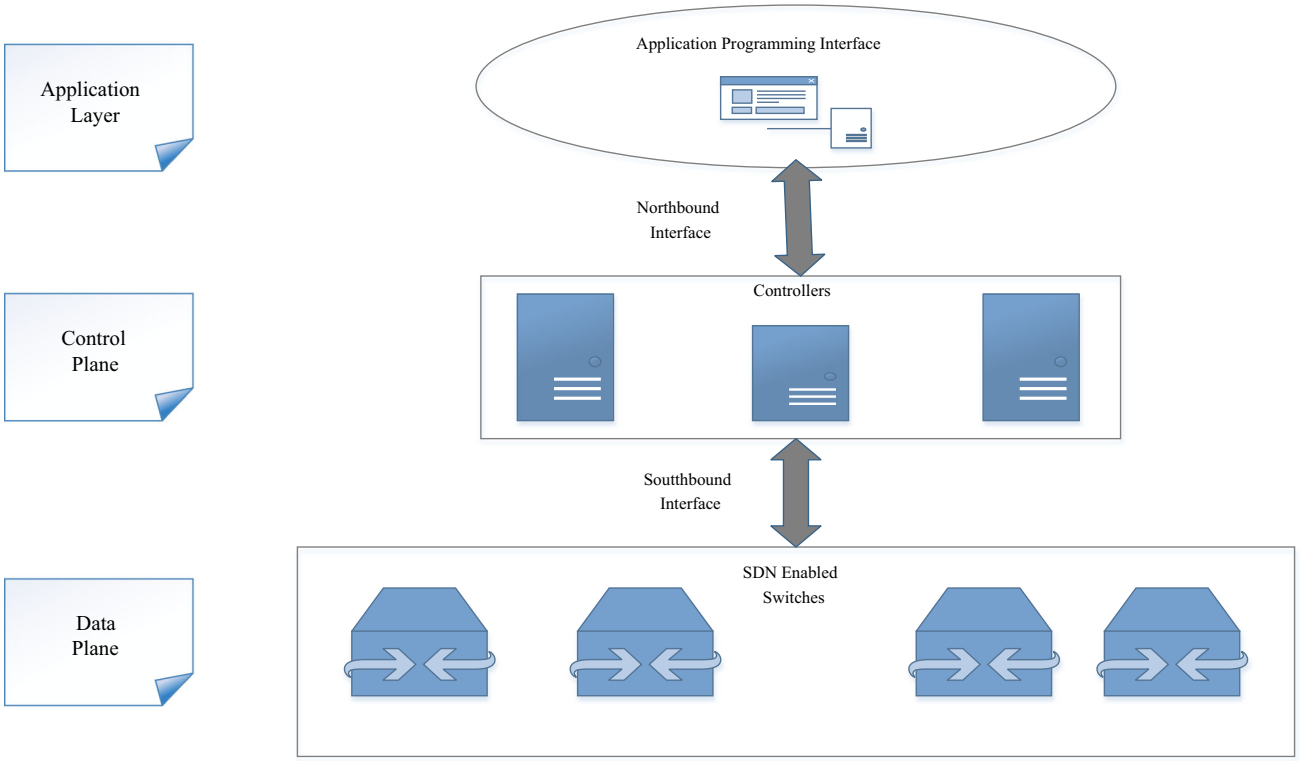

Fig. 1 SDN architecture

packet. SDN is different from traditional networks because it decouples the data and control planes and also makes networks programmable. The software applications can be programmed to make the network behave in the desired way. With the separation of planes and programmability feature, the SDN has made it easy to configure, control, monitor, safeguard and manage the networks.

The software-based analysis and control of traffic by SDN can be utilized by IoT to achieve an optimum security and traffic management. SDN incurs a lower cost and provides a global view of the network. In SDN architecture, there are customized applications programmed to control and manage the traffic. This feature can be used in an SDN-based IoT to manage the huge influx of data from various IoT domains. The programmability feature of SDN can also be utilized to enhance the security of IoT [30]. In this paper, the features of the SDN have been used for detection and alleviation of DDoS in IoT.

SDN-Based Detection and Alleviation of DDoS in IoT (SDIoT-DDOS-DA)

Most of the current DDoS attack detection, prevention and mitigation procedures in an IoT are deployed on the IoT network directly [31, 32]. Such strategies against DDoS in the IoT are resource consuming and might disable the IoT network in case of a huge DDoS attack, likes of which have surfaced recently. A generalized idea of an SDN-based IoT system is illustrated in Fig. 2. The centralized control can be used to achieve a better DDoS mechanism in the IoT. An SDN-based approach has been used in analyzing the traffic coming from and going to IoT. The traffic passes through an SDN-enabled switch.

The SDN acts as a gateway to the IoT network and determines whether the traffic is affected or not. The traffic patterns are compared against the predetermined patterns to find out the anomaly. A novel mechanism against DDoS called SDIoT-DDoS-DA is introduced in this paper. The proposed method has been implemented using SDN-WISE [33, 34]. SDN-WISE has been devised to provide an SDN-based stateful solution for Internet of Things or wireless sensor networks (WSN). SDN-WISE uses the SDN model in IoT or WSN.

Proposed Strategy

The proposed method against DDoS in IoT consists of the following modules: attack detection, identification and attack alleviation, respectively. These modules work in coordination and are implemented in the control plane. In order to detect and mitigate a DDoS attack, the system goes through various phases. Within the system, the phases are

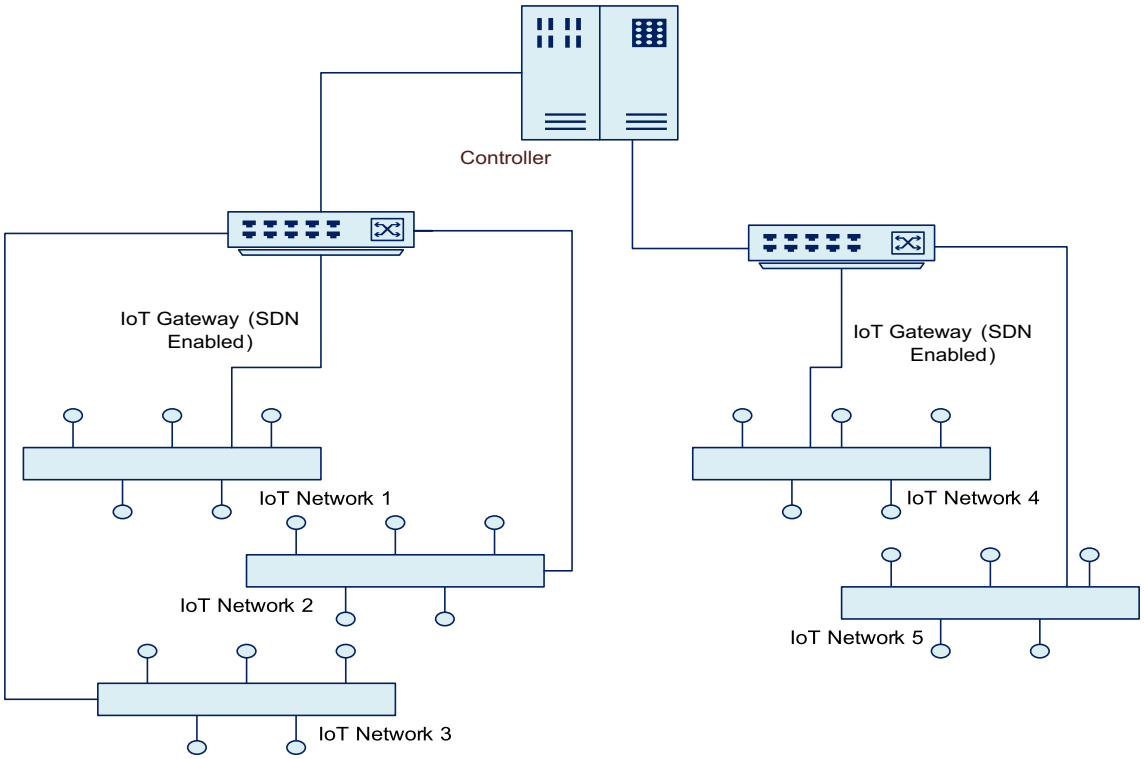

Fig. 2 Concept of SDN-based IoT architecture

changed as per the occurrence of events in the IoT network. The working of SDIoT-DDoS-DA is illustrated in Fig. 3.

Before starting SDIoT-DDoS-DA, the entire network is said to be in the *Normal Phase*. Once there is an increase in the flow of messages, the network is suspected to be under an attack and the system enters the *Detection Phase*. Once the system has entered into the *Detection Phase*, it has to find out whether the network is under the DDoS attack. The *Detection Phase* is activated when the increasing number of messages reaches a predetermined Threshold. If the system is found to be under DDoS attack, the attack *Identification Phase* starts. In this phase, the system tries to identify the attack path and originator of the attack. After identifying the attack source, the system shifts to the *Alleviation Phase*. In this phase, all traffic coming from the attack source is stopped. The transformation of the system through various phases is shown in Fig. 3. Each arrow represents the events which allow the system to navigate from one phase to another. In the *Alleviation Phase*, a mitigation strategy is implemented that aims to stop the attack traffic. The proposed mechanism against DDoS in IoT consists of various components used to carry out the work of various phases. The components are as follows:

The Normal Phase can recognize any variance from the usual behavior of the network; it does so by observing the frequency and volume of traffic. If it senses some abrupt increase in frequency and volume of messages that are trying to hit the IoT gateway, it passes on control to the *Detection Phase*. The *Detection Phase* contains a monitoring component which detects DDoS in the IoT network. When the control is passed to the *Detection Phase*, the monitoring component detects an anomaly and confirms DDoS attack. The system then shifts to the *Identification Phase*. The *Identification Phase* traces the attack path and locates the attacker by assessing the information from *Detection Phase* and by using the global view of the SDN. If the attacker is not identified, the system goes back to the previous phase. The *Alleviation Phase* is started after locating the attacker. In the *Alleviation Phase*, a suitable defense strategy is used to stop attack traffic. On recovering from the attack, the system shifts back to the Normal Phase. Each of the phases has been explained in detail in the following subsections.

Attack Detection Phase

In any DDoS defense strategy, the detection module is the key subsystem because it determines how proactive the system is. DDoS attack detection has been the major focus of recent research because DDoS attacks are escalating at a greater speed. The techniques for DDoS detection have been created mostly using statistical, data mining, machine learning, soft computing or knowledge-based methods [35].

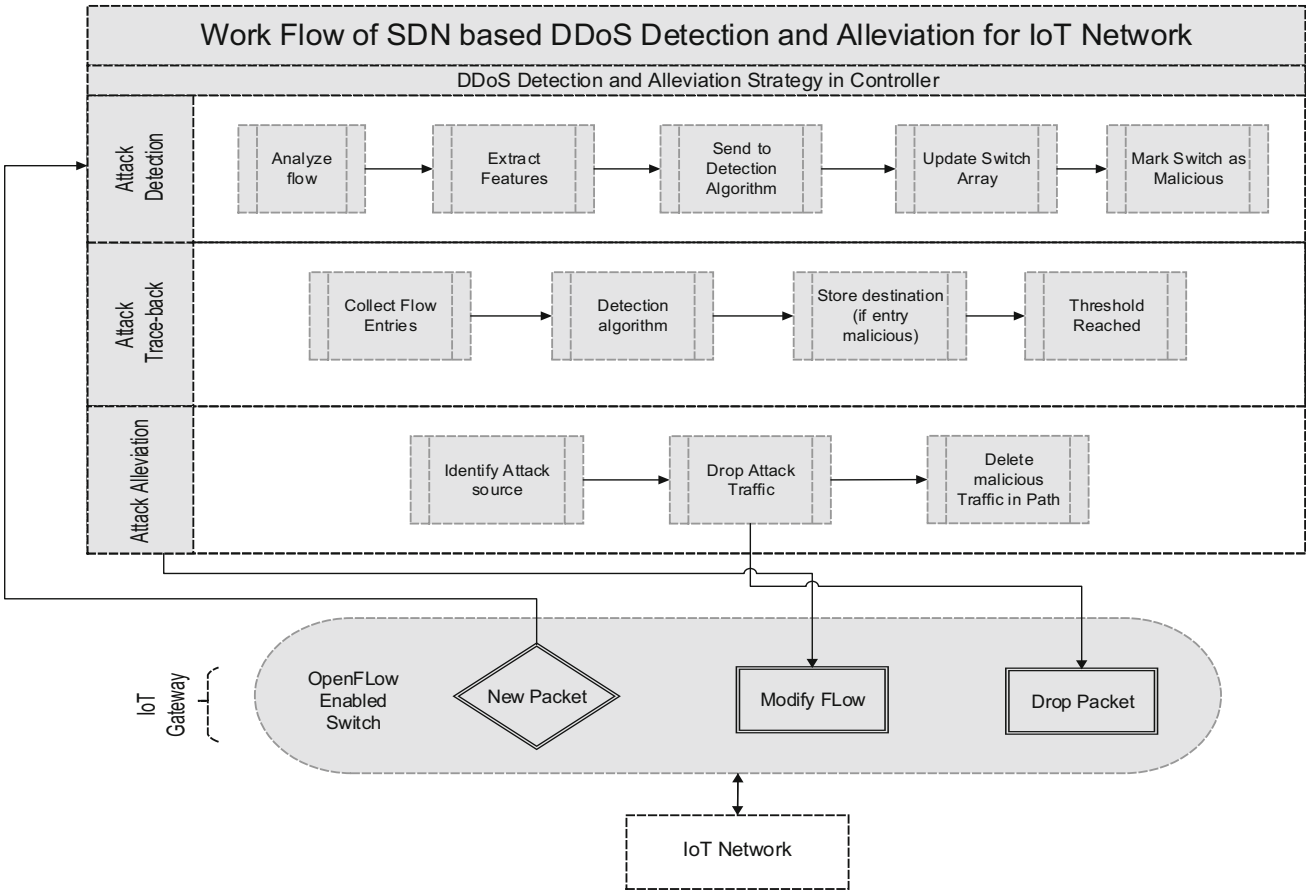

Fig. 3 Working of SDIoT-DDoS-DA

A DDoS detection mechanism includes a monitoring component that observes the network for any variance from normal behavior and then checks whether or not the deviation from normal is because of DDoS. If it detects a DDoS attack, it alerts the system or network administrator. The Detection Phase of SDIoT-DDoS-DA has two sub-modules: One sub-module monitors the system and discerns the anomalous flow of messages; another sub-module assesses the unusual behavior and confirms the DDoS attack. In the first sub-module, the rate at which messages are hitting the IoT gateway is calculated and the data stream abnormal detection algorithm is used to detect the outlier of messages [36]. Micro-Cluster Outlier Detection (MCOD) [37] has been used as the outlier detection algorithm. MCOD utilizes minimum CPU time among other popular data stream outlier detection algorithms. MCOD eliminates the need for range queries by storing the neighboring data points in micro-clusters.

The detailed process of the DDoS detection is depicted in Algorithms I and II. When a new message (n) arrives, the number of new messages termed as Counter (i) is increased by one. The modular division of Counter and Threshold (m), a predetermined maximum limit for the number of the new messages, is calculated. If the remainder is not zero, the new message is sent to the controller which handles it. Otherwise, the current time is noted. The time elapsed (t) is calculated by finding the difference between t_{curr} (current time) and t_{prev} (last time when remainder for the modular division of Counter and Threshold was zero). The Rate (u) of the new message is calculated by dividing Threshold with the time elapsed. The Rate is examined for abnormality using MCOD algorithm. If the Rate is outlier or abnormal, the second sub-module detects whether the abnormality is because of the DDoS attack, as explained in Algorithm II. Otherwise, the network controller is notified to handle the new message.

Algorithm I: Abnormality Monitoring

Input: new message = n
Output: abnormal behavior detection
Step 1: increase the Counter by one $:= i++$
Step 2: ***if*** $i\%m = 0$ ***then:***
Step 3: $t = t_{curr} - t_{prev}$
Step 4: $u = \frac{m}{t}$
Step 5: ***else*** send new message to controller/flow-visor
Step 6: ***end if***

Step 7: u (from Step 4) input to MCOD.
Step 8: ***if*** u is normal ***then.***
Step 9: notify the controller or flow-visor
Step 10: ***else*** find out whether the abnormality is because of DDoS (Algorithm II)

The time and space complexity for MCOD is given as [38]:

$$\underset{\text{Time Complexity}}{O((1-c)W\log(1-c)W) + kWlogk)}$$
$$\underset{\text{Space Complexity}}{O(cW + (1-c)kW)}$$

where $0 \leq c \leq 1$ denote the fraction of the window stored in micro-clusters, k is the count Threshold, and W is the window size.

MCOD eliminates the need for range queries by storing the neighboring data points in micro-clusters. Each micro-cluster has minimum $k + 1$ data points, where k is the count Threshold. One data point is taken as the center of the micro-cluster and has a radius equal to $R/2$, where R is the Threshold distance. Every data point in a micro-cluster is an inlier as per the triangular inequality. The data points that do not fall into any micro-clusters are stored in a list called PD (the list of data points that are not in micro-clusters). One list called event queue stores inliers that do not fall in of any clusters. The data points in PD with less than k neighbors are identified as outliers after the new slide and expired slide are processed in MCOD. MCOD eliminates the pair-wise distance computations and range queries and also requires lesser memory.

The *Detection Phase* has to analyze the abnormality precisely, to find out whether the outlier identified is because of DDoS attack. The remaining part of the *Detection Phase* is explained with the help of Algorithm II. Artificial neural networks are a preferred approach for efficient attack detection. The detection mechanisms based on neural networks can differentiate benign flow and malicious flow entry with higher accuracy. In SDIoT-DDoS-DA, multilayer perceptron (MLP) is used to detect the DDoS attack. MLPs are capable of getting required details from incomplete or complicated data which can be used to extract patterns and detect trends.

After the anomaly has been detected by the monitoring sub-module, the information from the flow entries is extracted from the controller and directed to the trained neural network or MLP. The MLP determines whether the traffic is ill-natured and DDoS based. Any neural network model needs to be trained before using it for real-time detection. The training is done using a dataset which is created in advance using characteristics of the malicious traffic. Within a dataset, a different set of values is used to represent malicious and benign traffic. The dataset is formed by mixing the characteristics of traffic and the values. The training of the neural network begins upon the initiation of the system. The features of the malicious and benign traffic are used as input to the MLP, and the values are the output [36]. These values are compared with the anomaly found, which helps in detecting a DDoS attack. The features input to MLP are: packet count matched by every flow entry, flow entry time and the rate of each flow entry. The features mentioned can vary depending on the accuracy to be achieved and are taken from the flow statistics of the controller. The eigenvalues for the MLP are created using these features which help in differentiating between benign and malicious traffic. The MLP used has one input layer, two hidden layers and one output layer. The number of perceptrons in the input layer is seven, the number of perceptrons in the hidden layer is fourteen, and the number of perceptrons in the output layer is one. The result of the MLP is stored in a list to be used by the identification module. Upon detection of a malicious flow entry, the destination address is determined and stored in a list called *malicious_ip_list*. If malicious flow entries increase and reach a Threshold value, a DDoS alert is raised, and the controller stops the processing of flow statistics message. The next flow entry is processed if the flow entry is benign and the number of malicious entries has not reached the maximum limit.

Algorithm II: Traffic Classification

Input: Flow statistics
Output: Identification of DDoS
Step 1: Extract features of the traffic from flow statistics.
Step 2: Classify the traffic using its features with MLP.
Step 3: Store the result of MLP or classify traffic in an array called attack_list to be used in *Identification Phase*.
Step 4: on detection of malicious flow entry, determine destination IP address, and store in a separate list called *malicious_ip_list.*
Step 5: ***if*** the number of entries in *malicious_ip_list* reaches the predefined threshold *then*
Step 6: raise DDoS attack alert.
Step 7: halt the processing of flow statistics message.
Step 8: search the *malicious_ip_list* and note the address with maximum occurrences.
Step 9: ***else*** process another flow entry.
Step 10: shift to *Identification Phase* of the system

Attack Identification Phase

In the *Detection Phase*, the result of the MLP is stored in a list called *attack_list* and sent to the *Identification Phase*. The attack source is identified by analyzing the results from *Detection Phase* and the network topology. The *Identification Phase* includes an identification module that makes uses of the MLP model from the *Detection Phase* to

determine network devices lying in the attack path. Based on the content of the malicious traffic found in the IoT gateway, the gateway is labeled as infected. If the proportion of malicious flow entries in the IoT gateway is lesser than a predetermined value, then the gateway is termed as non-infected. The attacked gateway and the attack path are identified accurately by SDIoT-DDoS-DA because of the global view of the network provided by an SDN controller.

Attack Alleviation Phase

The *Alleviation Phase* in SDIoT-DDoS-DA is the final phase of the system that extenuates the DDoS attack detected in previous phases. It prevents the network from further worsening and restores it to a normal state. The *Alleviation Phase* acts as a response system against the DDoS attack detected, and it starts after attack path and the attack origin have been traced. The attack *Alleviation Phase* includes the alleviation module that drops the traffic from attack source. The traffic from attack source device is blocked by inserting a high-priority flow table of the attack origin device. Such high-priority flow entries are known as blocking traffic. When the attack traffic tries to leave the attack source device, the attack traffic is matched to high-priority flow entries in the table. Based on the matching of attack traffic with blocking flow, it gets dropped; hence, the attack is stopped.

Performance Evaluation

This section assesses the performance of proposed mechanism against DDoS in the IoT environment. The proposed mechanism is compared with few other similar DDoS defense approaches at the end of this section. The proposed system is implemented using the SDN-WISE framework. The controller used is Open Network Operating System (ONOS), and the DDoS attack traffic is generated using Trinoo. Trinoo is one of the famous DDoS attack tools widely used to attack several famous sites. Trinoo produces UDP floods attack and uses TCP between attacker and control master program [39]. SDN-WISE is based on Mininet [40] which is a standard tool used to simulate SDN. To simulate the DDoS attack, packet records of DDoS are taken in test bed and replayed. During the attack, request rate on the IoT gateway (Fig. 4) increases considerably.

The experimental setup consists of a network having eight switches, twenty hosts and twenty-five devices. The attack has been launched using Trinoo. The attack originates from five hosts that try to attack the host whose IP is

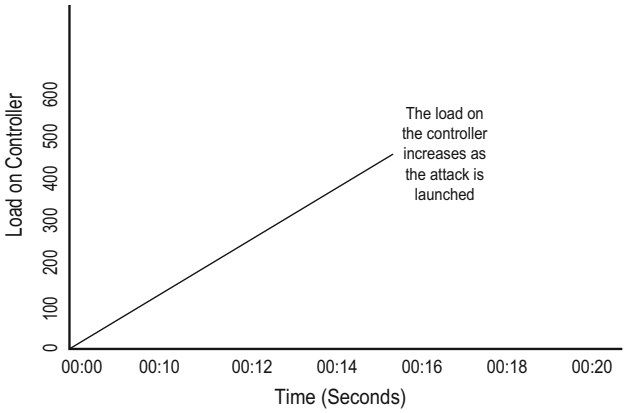

Fig. 4 Load on controller during DDoS Attack

10.0.0.9. The simulation start time along with activation of each module of the proposed system is shown in Fig. 5.

At the beginning of the experiment, the system is in the initial state where the MLP model is trained. The system starts at 14:09:05. Upon sensing an increased rate of messages, the SDIoT-DDoS-DA enters the *Detection Phase* between 14:09:15 and 14:09:20. The DDoS attack is found at 14:09:20, and then, the system enters the *Identification Phase* at 14:09:21. The *Alleviation Phase* is started subsequently which drops the malicious traffic. The results are shown in Fig. 6a, b which depict the impact of the DDoS attack on detection of malicious traffic as False Positives and False Negatives. Figure 6a shows the False Negative errors caused when the DDoS attack is launched. The False Negative errors predominantly occur before the attack is launched between 14:09:00 and 14:09:06. Once the DDoS attack rate increases, the False Negative Errors are reduced. As shown in Fig. 6b, the occurrence of False Positive errors increases with rising attack rate between 14:09:06 and 14:09:20. During the attack, the normal traffic adds to the increasing request rate and hence there

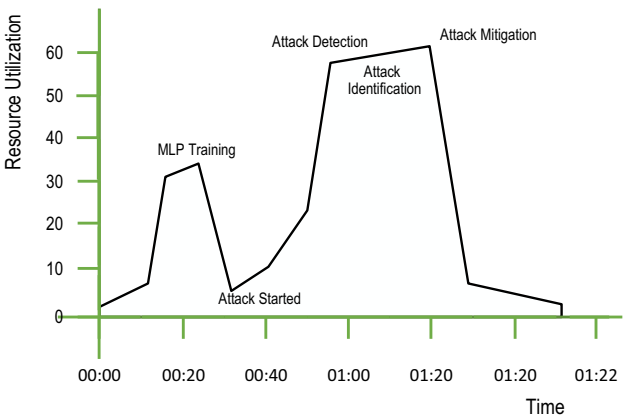

Fig. 5 Resource utilization at different phases of SDIoT-DDoS-DA

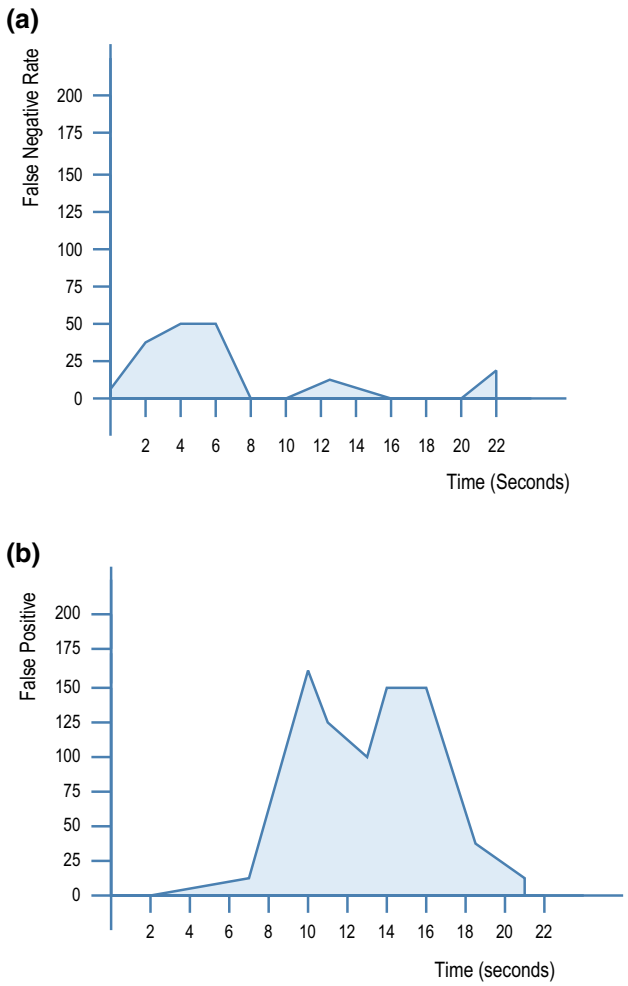

Fig. 6 **a** The False Negative results for SDIoT-DDoS-DA. **b** The False Positive results for SDIoT-DDoS-DA

are more False Positive errors. Once the attack alleviation starts, the errors are reduced considerably.

The *Alleviation Phase* drops the infected traffic after the attack is detected and confirmed in *Detection Phase*. The results from the above experiment show that soon after the DDoS attack was launched, the abnormality was identified by the monitoring sub-module of the *Detection Phase*. The monitoring sub-module starts analyzing the deviation to find out whether it is a DDoS or not. As soon as DDoS is detected, the attack trail is traced by identification component of *Identification Phase*. The identification component finds the attack path and locates the attack source. The alleviation component blocks the DDoS attack traffic.

Results and Discussion

During the attack simulation, the IoT gateway received requests from the attack source as well as the non-attack hosts. The *Detection Phase* of SDIoT-DDoS-DA started when the number of packets hitting the IoT gateway increased. It collected all the traffic of targeted host under examination. The captured traffic logged 1054 requests between the start and end of the test. The information regarding the source and destination IP addresses and ports of each request is also logged. Since non-attack hosts and five emulated attack hosts were known, the logged information was used for evaluation. The results obtained from the simulation showed that the system classified 876 requests as illegitimate access, out of which 11 turned out to be False Positives. The detection module also predicted 178 commands to be legitimate access requests, of which 32 were False Negative. This information has been summarized in terms of a confusion matrix with respect to the illegitimate requests (Table 1).

Various accuracy measures can be calculated from the confusion matrix. These measures are listed in Table 2.

However, for comparative evaluation, the performance of the SDIoT-DDoS-DA has been appraised in terms of *Positive Production Power* (PPP) and *Sensitivity*. Tamotsu Kawamura et al. [22] have used the terms *Precision* and *Recall* to refer to *PPP* and *Sensitivity,* respectively. The authors [16] have referred to *Sensitivity* by *Detection Rate (DR)* as well.

As can be seen from Table 2, *Positive Productive Power/Precision*, given by $\boldsymbol{TP}/(\boldsymbol{TP}+\boldsymbol{FP})$, and *Sensitivity/Recall/Detection Rate*, given by $\boldsymbol{TP}/(\boldsymbol{TP}+\boldsymbol{FN})$, are valued at 0.9874 and 0.9643, respectively. The authors [22] on the other hand have a *PPP/Precision* of only 0.92, which is much less compared to our system. However, [22] has a perfect *Sensitivity/Recall/DR* value of 1, compared to 0.9643 of our system. The comparison of the two systems is reported in Table 3.

Additionally, the performance of the proposed system has also been evaluated in terms of *Sensitivity/Recall/DR* and False Positive Rate/*False Alarm Rate FPR/FAR*, for the purpose of comparison with MOPF [16], which has

Table 1 Confusion matrix (performance of SDIoT-DDoS-DA)

Request type	Detected as illegitimate	Detected as legitimate
Illegitimate	865	11
Legitimate	32	146
N = 1054		

Table 2 Accuracy measures for performance of SDIoT-DDOS-DA

Measure	Calculation	Values
Prevalence	$\frac{TP+FN}{N}$	0.851
Overall diagnostic power	$\frac{FP+TN}{N}$	0.149
Correct classification rate	$\frac{TP+TN}{N}$	0.9592
Sensitivity	$\frac{TP}{TP+FN}$	0.9643
Specificity	$\frac{TN}{FP+TN}$	0.9299
False Positive Rate	$\frac{FP}{FP+TN}$	0.0701
False Negative Rate	$\frac{FN}{TP+FN}$	0.0357
Positive predictive power	$\frac{TP}{TP+FP}$	0.9874
Negative predictive power	$\frac{TN}{FN+TN}$	0.8202
Misclassification rate	$\frac{FP+FN}{N}$	0.0408
Odds ratio	$\frac{TP*TN}{FN*FP}$	358.7784
Kappa	$\frac{(TP+TN)-\left(\frac{(TP+FN)(TP+FP)+(FP+TN)(FN+TN)}{N}\right)}{N-\left(\frac{(TP+FN)(TP+FP)+(FP+TN)(FN+TN)}{N}\right)}$	0.8475
Normalized mutual information (NMI) n(s)	$1-(-TP.\ln(TP)-FP.\ln(FP)-FN.\ln(FN)-TN.\ln(TN)+(TP+FP).\ln(TP+FP)+(FN+TN).\ln(FN+TN))/(N.\ln(N)-((TP+FN).\ln(TP+FN)+(FP+TN).\ln(FP+TN)))$	0.6778

Table 3 Comparison of SDIoT-DDOS-DA with NTP method [22]

	PPP/Precision	Sensitivity/Recall/DR
NPT	0.92	1
SDIoT-DDoS-DA	0.9874	0.9643

Table 4 Comparison of SDIoT-DDoS-DA with MOPF [16]

Detection mechanism	DR (%)	FAR (%)
MOPF	80.95	5.92
SDIoT-DDoS-DA	96.4325%	7.01

used *FPR/FAR* as an accuracy evaluation metric. For a binary classifier,

$$\text{False Positive Rate/False Alarm Rate} = (1 - \text{Specificity}) = (\text{false detections})/(\text{all detections})$$

and is given by $FPR/FAR = FP/(TN + FP)$.

The performance of SDIoT-DDoS-DA for detecting DDoS attacks in IoT in terms of *Sensitivity/Recall/DR* and *False Alarm Rate* FAR is given as:

$$\text{Sensitivity/Recall/DR} = 96.4325\% \text{ and } \text{FAR} = 7.01\%.$$

For evaluating the performance of the proposed model, the proposed detection module was compared with MOPF [16]. The results of this comparison are reported in Table 4. As seen in Table 4, SDIoT-DDoS-DA offers better Detection Rate. However, our system has a higher saturation of False Alarm Rate.

Table 5 Comparison of SDIoT-DDoS-DA with works in [21, 41]

	PPP/Precision	Sensitivity/Recall/DR
SDIoT-DDoS-DA	98.74	96.43
IoT-New	86.32	–
IoT-IDM	98.53	95.94

The Precision rate of our monitoring method is 98.74%, while the Precision rate of the method proposed in IoT-New [21] is 86.32% which is 13.42% lesser than SDIoT-DDoS-DA. The comparison is depicted in Table 5. The experimental results of IoT-IDM [41] showed a Precision rate of 98.53% and a Recall rate of 95.94%, while the Precision and Recall rates of SDIoT-DDoS-DA are 98.74% and 96.43%. The results are slightly lesser than the proposed method. The difference in the values is depicted in Table 5.

Conclusion and Future Work

The IoT is expanding, and its presence is felt in every field. Apart from inheriting the security and privacy issues from the Internet, IoT has been a great aid for hackers who aim to create disastrous cyber-attacks. Intermittent DDoS attacks of huge capacity are one of the major threats that have resulted in the growth of IoT. A robust and flexible security mechanism to abate DDoS in IoT is indispensable. This paper discusses the impact of DDoS attack in IoT and

introduces a flexible SDN-based novel method for detecting and mitigating DDoS. SDN offers improved network control and defines a novel way of data transfer by the decoupling of control and data planes. The initial tests are performed on a limited dataset which can be extended for a larger volume of attack. The future work can be the inclusion of DDoS prevention in IoT networks and the implementation of the simulation work on real IoT hardware. A strict authentication mechanism can be proposed to prevent IoT devices from turning into botnets.

Acknowledgements The authors wish to thank MANF UGC, Govt. of India, for providing financial support under MANF-UGC (MANF-2015-17-JAM-60506) program to carry out this work.

References

1. A. Rayes, S. Salam, *Internet of Things-from Hype to Reality: The Road to Digitization* (2016)
2. I. Yaqoob et al., Internet of things architecture: recent advances, taxonomy, requirements, and open challenges. IEEE Wirel. Commun. **24**(3), 10–16 (2017)
3. L. Atzori, A. Iera, G. Morabito, From 'smart objects' to 'social objects': the next evolutionary step of the internet of things. IEEE Commun. Mag. **52**(1), 97–105 (2014)
4. A. Remke, B.R. Haverkort, *Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems*, vol. 9629 (2016), pp. 1–4
5. M. De Donno, N. Dragoni, A. Giaretta, A. Spognardi, *Analysis of DDoS-Capable IoT Malwares*, vol. 11 (2017), pp. 807–816
6. J. Wei, *DDoS on Internet of Things—a Big Alarm for the Future* (2016)
7. P. Paganini, 150,000 IoT Devices behind the 1Tbps DDoS attack on OVHSecurity Affairs, 2016-09-27 (2016), https://securityaffairs.co/wordpress/51726/cyber-crime/ovh-hit-botnet-iot.html. Accessed 27 Mar 2018
8. G. Corfield, Security man Krebs' website DDoS was powered by hacked Internet of Things botnet • The Register (2016), https://www.theregister.co.uk/2016/09/26/brian_krebs_site_ddos_was_powered_by_hacked_internet_of_things_botnet. Accessed 27 Mar 2018
9. J. Malik, Threats Converge: IoT Meets Ransomware (2017). https://www.darkreading.com/vulnerabilities—threats/threats-converge-iot-meets-ransomware/a/d-id/1328304? Accessed 27 Mar 2018
10. Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, D. Qiu, Security of the Internet of Things: perspectives and challenges. Wirel. Net.works **20**(8), 2481–2501 (2014)
11. W. Azka, S. Revathi, Protocols for Secure Internet of Things. Int. J. Educ. Manag. Eng. **7**(2), 20–29 (2017)
12. P. Goransson, C. Black, T. Culver, *Software Defined Networks: A Comprehensive Approach* (2016)
13. M.C. Dacier, H. Konig, R. Cwalinski, F. Kargl, S. Dietrich, Security challenges and opportunities of software-defined networking. IEEE Secur. Priv. **15**(2), 96–100 (2017)
14. J. Kim et al., Standard-based IoT platforms interworking: Implementation, experiences, and lessons learned. IEEE Commun. Mag. **54**(7), 48–54 (2016)
15. Á.L. Valdivieso Caraguay, A. Benito Peral, L.I. Barona López, L.J. García Villalba, SDN: evolution and opportunities in the development IoT applications. Int. J. Distrib. Sens. Netw. **10**, 735142 (2014)
16. M. Sheikhan, H. Bostani, A hybrid Intrusion Detection System for Internet of Things, in *8th Symp. Telecommun.*, no. 3 (2016), pp. 2395–4396
17. O. Salman, I. Elhajj, A. Chehab, A. Kayssi, Software Defined IoT security framework, in *2017 4th Int. Conf. Softw. Defin. Syst. SDS 2017* (2017), pp. 75–80
18. M. Miettinen et al., IoT sentinel demo: automated device-type identification for security enforcement in IoT, in *Proc. - Int. Conf. Distrib. Comput. Syst.* (2017), pp. 2511–2514
19. P.K. Sharma, S. Singh, Y.S. Jeong, J.H. Park, DistBlockNet: a distributed blockchains-based secure SDN architecture for IoT networks. IEEE Commun. Mag. **55**(9), 78–85 (2017)
20. C. Li, Z. Qin, E. Novak, Q. Li, Securing SDN infrastructure of IoT-Fog networks from MitM attacks. IEEE Internet Things J. **4**(5), 1156–1164 (2017)
21. T. Xu, D. Gao, P. Dong, H. Zhang, C.H. Foh, H.C. Chao, Defending against new-flow attack in SDN-based Internet of Things. IEEE Access **5**, 3431–3443 (2017)
22. T. Kawamura, M. Fukushi, Y. Hirano, Y. Fujita, Y. Hamamoto, An NTP-based detection module for DDoS attacks on IoT, in *2017 IEEE Int. Conf. Consum. Electron. - Taiwan, ICCE-TW 2017* (2017), pp. 15–16
23. Y.M.P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, C. Rossow, IoTPOT: a novel honeypot for revealing current IoT threats. J. Inf. Process. **24**(3), 522–533 (2016)
24. S.D. Odabasi, M.S. Haskırış, Internet of Things (IoT), security and Distributed Denial of Service (DDoS) attack, in *1st Int. Mediterr. Sci. Eng. Congr. (IMSEC-2016); Congr. Center, Çukurova Univ. Adana, Turkey*, no. October 2016 (2016), pp. 4934–4938
25. P. Bull, R. Austin, E. Popov, M. Sharma, R. Watson, Flow based security for IoT devices using an SDN gateway, in *Proc. - 2016 IEEE 4th Int. Conf. Futur. Internet Things Cloud, FiCloud 2016* (2016), pp. 157–163
26. M. De Donno, N. Dragoni, A. Giaretta, M. Mazzara, AntibIoTic: protecting IoT devices against DDoS attacks. Adv. Intell. Syst. Comput. **717**, 59–72 (2018)
27. C. Zhang, R. Green, Communication security in Internet of Thing: preventive measure and avoid DDoS attack over IoT network, in *Proc. 18th Symp. Commun. Netw.*, no. January 2015 (2015), pp. 8–15
28. W. Azka, S. Revathi, A. Geetha, *A Survey of Applications and Security Issues in Software Defined Networking*, no. March (2017), pp. 21–28
29. SDN/OpenFlow|Flowgrammable: https://flowgrammable.org/sdn/openflow/ (2015). Accessed 27 Mar 2018
30. S.K. Tayyaba, M.A. Shah, O.A. Khan, A.W. Ahmed, Software Defined Network (SDN) based Internet of Things (IoT): a road ahead, in *Proc. Int. Conf. Futur. Networks Distrib. Syst.* (2017), pp. 15:1–15:8
31. P.C. Vinh, V. Alagar, Context-aware systems and applications: 4th international conference, ICCASA 2015 Vung Tau, Vietnam, November 26–27, 2015 revised selected papers 123, *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 165 (2016), pp. 62–72
32. P. Kasinathan, C. Pastrone, M.A. Spirito, M. Vinkovits, Denial-of-Service detection in 6LoWPAN based Internet of Things, in *Int. Conf. Wirel. Mob. Comput. Netw. Commun.* (2013), pp. 600–607
33. L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. Proc. IEEE INFOCOM **26**, 513–521 (2015)

34. A.C.G. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, Towards a software-defined Network Operating System for the IoT, in *IEEE World Forum Internet Things, WF-IoT 2015—Proc.* (2015), pp. 579–584
35. D.K. Bhattacharyya, J.K. Kalita, *DDoS ATTACKS*, 1st edn. (CRC Press, New York, 2016)
36. Y. Cui et al., SD-Anti-DDoS: fast and efficient DDoS defense in software-defined networks. J. Netw. Comput. Appl. **68**, 65–79 (2016)
37. D. Georgiadis, M. Kontaki, A. Gounaris, A. Papadopoulos, K. Tsichlas, Y. Manolopoulos, Continuous outlier detection in data streams: an extensible framework and state-of-the-art algorithms, in *Proc. 2013 ACM SIGMOD Int. Conf. Manag. Data* (2013), pp. 1061–1064
38. L. Tran, L. Fan, C. Shahabi, Distance-based outlier detection in data streams. Proc. VLDB Endow. **9**(12), 1089–1100 (2016)
39. P. Boyle, *Distributed Denial of Service Attack Tools: Trinoo and Wintrinoo* (2011)
40. M. Team, Mininet An Instant Virtual Network on your Laptop (or other PC). https://mininet.org/
41. M. Nobakht, *A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow* (2016), pp. 147–156