Research Article

S-DPS: An SDN-Based DDoS Protection System for Smart Grids

Hassan Mahmood,^{1,2} Danish Mahmood^{1,2}, Qaisar Shaheen³, Rizwan Akhtar,⁴ and Wang Changda¹

¹School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China

²Department of Computer Science, SZABIST, Islamabad, Pakistan

³Department of Computer Science, Superior College, Lahore, Pakistan

⁴Department of IT and Computer Science, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan

Correspondence should be addressed to Danish Mahmood; dr.danish@szabist-isb.edu.pk and Wang Changda; changda@ujs.edu.cn

Received 31 December 2020; Revised 19 February 2021; Accepted 8 March 2021; Published 22 March 2021

Academic Editor: Shehzad Chaudhry

Copyright © 2021 Hassan Mahmood et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Information Communication Technology (ICT) environment in traditional power grids makes detection and mitigation of DDoS attacks more challenging. Existing security technologies, besides their efficiency, are not adequate to cater to DDoS security in Smart Grids (SGs) due to highly distributed and dynamic network environments. Recently, emerging Software Defined Networking- (SDN-) based approaches are proposed by researchers for SG's DDoS protection; however, they are only able to protect against flooding attacks and are dependent on static thresholds. The proposed approach, i.e., Software Defined Networking-based DDoS Protection System (S-DPS), is efficiently addressing these issues by employing light-weight Tsallis entropy-based defense mechanisms using SDN environment. It provides early detection mechanism with mitigation of anomaly in real time. The approach offers the best deployment location of defense mechanism due to the centralized control of network. Moreover, the employment of a dynamic threshold mechanism is making detection process adaptive to the changing network conditions. S-DPS has demonstrated its effectiveness and efficiency in terms of Detection Rate (DR) and minimal CPU/RAM utilization, considering DDoS protection focusing smurf attacks, socket stress attacks, and SYN flood attacks.

1. Introduction

There is a drastic increase in energy dependence from very minute to huge activity especially the cloud-based data centers and Internet of Things (IoT), which have a dire need of availability, reliability, and efficiency of power systems. This requirement paved path towards the Smart Grid (SG) paradigm that ensures two-way communication within power systems. It holds the capability to remove the constraints of a traditional grid infrastructure and provide power systems that are scalable, dynamic, situation-aware, and flexible. On the other hand, such facilities give birth to complexity, heterogeneity, and interconnectivity of diverse ICT requirements due to which the existing network paradigms and security strategies are marked as ineffective

[1, 2]. Moreover, the IP-enabled communication infrastructure in SGs raises the likelihood of malicious activities and attacks. Such attacks may result in wrong smart meter readings or incorrect demands or responses to or from electricity company or they can be severe for power generation systems [3].

Millions of consumers are serviced by SG. The service provided by SG is crucial and the availability of such a service is extremely important. The SG makes up a cyber-physical system (CPS) and a single error in any part of the system can lead to a direct or indirect catastrophic effect on human life [4]. Distributed Denial of Service (DDoS) attacks are also making more frequent appearances and are becoming more sophisticated and severe because of the fact that the existing protection mechanisms are not capable to deal with such threats. Hence,

detection, mitigation, and prevention of DDoS attacks are now on the top most priority of engineering industries and research arena. Researches have come up with SDN-based approaches to handle DDoS attacks in SG [5]. However, still experimental validation of the proposed approaches is lagging. Also, SGs are safeguarded against High-Rate (HR)-DDoS attacks only during their detection and mitigation approach [1] [6]. This level of safety is not enough for a sensitive ICT infrastructure such as SG that carries mission critical information. Because of these reasons, there exists a desperate need for further research regarding SDN-based security protocols in SG to ensure a safe and light-weight mechanism against DDoS, having a capability to detect in the early stages and mitigation of varied level of DDoS attacks.

The remainder structure of the paper is organized as follows: Section 2 discusses related work with critical evaluation of literature, motivation with problem statement, and approach with contribution. Section 3 discusses system model with implementation constraints. Experimental setup with evaluation criteria is discussed in Section 4. Results and discussion are presented in Section 5. Section 6 discusses performance evaluation of the approach. Finally, Section 7 presents conclusion with future work.

2. Literature Review

Considering the wide spread of ICT and upcoming IoT devices, applications, and scenarios in almost every field of life, the authors in [7] showcased the vulnerabilities that may attract negative attentions. Moreover, the authors discussed state-of-the-art work regarding mitigation of such malicious activities. Researchers from academia and industry have shown interest and utilize new network paradigm, i.e., SDN to deal with underlying security risks in SG communication network [5]. The authors in [8] presented a comprehensive survey focusing SG communication security measures and privacy breaches. Major emphasis is given to privacy handling within SG communication networks. In [9], the authors presented a taxonomy of network attacks focusing fog-based smart grid SCADA systems. The authors in this study classify intrusion detection systems (IDSs) as major solution for the attacks; however, they focused mainly on machine learning approaches which at times are more time consuming and compel in comparison to entropy-based IDSs.

The authors in [10] used blockchain for securing the data and SDN to deal with control issues and scalability. Furthermore, the authors in [11] also used blockchain to secure energy sector, mainly authentication and privacy issues.

SDN is recently used in SG to provide a resilient SDN-based security framework/simulators and communication architecture. Researchers have utilized either single or multicontroller architecture to establish the underlying network infrastructure [12] which is considered to be the first SG-enabled simulator which is resilient and secured. The proposed security module is able to detect and resolve DoS attack within 60 seconds with no impact on bus system. However, maximum power capacity allowed on each bus/branch is not mentioned to address how much additional load other branches can bear in case of failure. Static threshold of 40% for number of packets/sec is used in

detection mechanism. Similarly, in [1], a novel SDN-based communication architecture for resiliency and security of microgrid operations is proposed. They have used three applications, i.e., self-healing mechanism, network verification, and intrusion detection. Self-healing mechanism uses rapid network configuration changes to mitigate further penetration by doing traffic isolation. Network verification is implemented using consistent updates feature, to avoid network instability by ensuring consistency of packets. Specification-based intrusion detection system is proposed; however, experimental validation is missing in the work.

Entropy-based approaches have been widely in traditional networks to provide DDoS protection. These approaches have proved to be useful in SDN environment as well, providing better detection efficiency [13]. The authors in [14] used open flow SDN controller to detect DDoS attacks on SG. As this is basic feature of SDN framework, the proposed methodology is also situation aware; however, for anomaly detection, an entropy-based mechanism is proposed which not only detects but also mitigates the attacks. However, the authors have not enhanced their proposed model to adoptively change according to situation and environment.

The authors in [15] presented a DDoS traceback mechanism under the umbrella of SDN architecture. The authors established an anomaly tree by analysing the communication flow changes via base station nodes. Once the anomaly tree is formulated, traceback scheme calls out any of different DDoS protection algorithms depending upon the nature and severeness of the attack. The authors claim that proposed scheme is better than the state-of-the-art frameworks regarding detection and trace back time with minimal usage of resources. In the future, the authors intend to optimize this approach by making it adoptive such that it can detect most types of the DDoS attacks.

A scheduling algorithm based on two levels is proposed in [16] to make sure better QoS regarding power services and communication network. For this purpose, an SDN controller is utilized and in first level, a scheduling mechanism is devised focusing priority in terms of delay constrained power services. Once priority-based services are schedules, then congestion and queueing control mechanism follows which ensures minimal delay with respect to the priority assigned. The authors used Mininet and Ryu controllers for simulation purposes. The proposed approach reduced delay and packet loss ratio with respect to state-of-the-art work. An elliptic curve cryptography (ECC) is presented in [17] and the proposed scheme which is based on mutual authentication by using biometric system. The authors claim to eliminate many authentication attacks.

Moreover, ECC technique supersedes other state-of-the-art protocols considering the performance metrics of communication and computational time considering SG environment. The authors in [18] proposed multilevel autoencoders-based IDS for DDoS attacks in SGs. The authors claim to have better accuracy in predictive analysis with other state-of-the-art methods. In [19], the authors presented a novel SDN-based IDS for SG. Basic feature of SDN, i.e., centralized controller in control

plane, is made distributed by using blockchain approach. The proposed model is simulated using AnyLogic and results declare it as more effective in terms of DDoS detection with state-of-the-art frameworks. Moreover, this approach also reduces the controller overhead; however, the delay in decision making is the trade-off that is not required in demand responsiveness feature of SG.

The authors in [20] present a novel entropy-based statistical approach in multicontroller SDN environment approach which is proposed for early detection and mitigation of DDoS attack. Apart from early detection, it is able to identify the attack path as well to apply the mitigation strategy instantly after detection. Shannon entropy with experimental static threshold against “DA” is used as detection mechanism, whereas Drop/block ports mechanism is used for mitigation purpose. Experimental validation for backup controller functionality, in case of primary controller failure, is missing. Threshold mechanism should have been adaptive rather than static, considering dynamic nature of modern networks. The authors have not addressed the efficacy of approach in protecting against LR-DDoS attacks. Further, approach should have been validated using performance metrics like DR and FPR.

Apart from SDN-based approaches for DDoS protection, it is important to discuss existing entropy-based approaches that have been successful in detecting DDoS attacks in traditional networks. Different variants of entropy are available for detecting DoS/DDoS attacks, i.e., generalized entropy, Tsallis entropy, and normalized entropy. Each of them can achieve varying DR and FPR for HR and LR-DDoS attacks. Some are able to detect both types of DDoS attacks with better DR and FPR, where some are best suited for a single type only. These solutions depend on the traffic features and perform statistical procedures on normal and attack traffic to do the comparison in order to find the anomaly. The authors in [21] present a generalized entropy-based feature selection technique which is used to detect network anomalies from real-life WAN traffic data with a high DR and low FPR. An outlier score function is used to detect the anomalies. The algorithm was evaluated against other techniques like LOF and ORCA using dataset Zoo. They achieved DR of 94.11% and FPR of 2.38%, higher than the other two approaches. However, user-defined parameters for threshold values are used. Although these values are set after conducting training on datasets, but still, it poses a limitation with respect to dynamic nature of networks and underlying attacks. The approach did not directly work on categorical and mixed types data.

A variant of Renyi entropy is proposed in [22], as a light-weight detection system utilizing extended entropy-based metric to detect HR-DDoS flooding attack and IP traceback. The proposed approach is evaluated against other entropy metrics like Shannon entropy and Kullback–Leibler divergence using both simulated and real-time DDoS datasets. Another important variant of parameterized entropy, i.e., Tsallis entropy, is utilized by researchers for anomaly detection. A feature-based Anomaly Detection System (ADS) using Tsallis entropy at device level is proposed in [23] and is capable of detecting and classifying known and unknown anomalies with additional information regarding network usage. Primitive properties of

flows like SA, DA, SP, and DP and derived flow properties at device and network level, i.e., out-degree, in-degree, per flow, per packet, per byte, packet per sample (pps), etc., are used in flow extraction process. Based on the discussion above, it can be observed that the authors have not highlighted the capability of their approach to detect both HR and LR-DDoS attacks. Moreover, static thresholds based on experiments are utilized in their approaches which cannot prevail in dynamic and complex environments like SG. Real dataset for DDoS attacks based on SG networks are not publically available easily [24]; hence, researchers have used simulated datasets for validation of their work.

Tsallis entropy metric has performed well, as per validation metrics, compared to other entropic metrics in detecting varying number of DDoS attacks, i.e., both LR-DDoS and HR-DDoS attacks. For effective DDoS defense mechanism, mitigation strategies should also be incorporated with intrusion detection system. Placement of detection mechanism is way important for efficient detection and in-time capitalization of DDoS attack. Such fact has not been addressed by many of the researchers. Finally, Tsallis entropy metric, besides its efficiency with respect to DR and FPR, has not been tested in an SDN environment. Utilizing SDN for securing SGs is in focus for energy engineering industries and research arena as well. The authors in [25] orchestrate a strategic connection, monitoring SDN controllers and sources of new flow requests that are threatening for DDoS attack. Compromised switches are identified and a noncooperative game is orchestrated using dynamic Bayesian network. The authors in [26] proposed a DDoS detection mechanism in SGs using Convolutional Neural Network. Variance fractal dimension trajectory is used as a preprocessing tool, whereas postprocessing of data is conducted by employing support vector machine. The authors claimed to achieve 87.35% accuracy in DDoS detection. Critical evaluation of literature is given in Tables 1 and 2.

2.1. Motivation and Problem Statement. In light of the above discussion, it can be concluded that SDN significantly addresses the deployment locality requirement to its centralized controller architecture. Further, entropy-based techniques used by the researchers rely on experimental-based thresholds and do not adapt to changing network conditions. Therefore, it necessitates developing an adaptive light-weight entropy-based defence mechanism using SDN environment for SG, providing early detection and mitigation of anomaly in real time. Real-time reconfiguration based on network conditions is required to change static thresholds and also to make it appropriate for high Detection Rate (DR) and low False Positive Rate (FPR). Here DR measures portion of the attacks that are detected correctly by the system and FPR provides the percentage of events that are reported as negative events where actually they are positive events. This factor makes it highly inappropriate for SG due to dynamic nature and heterogeneity.

With the advent of IoT, security concerns related to user and network resources have become even more critical and prone to attacks. SG being one important application of IoT also shares the same security threats that exist in traditional IoT environment. However, protection of DDoS attacks in

TABLE 1: SDN-based security approaches for DDoS protection.

Approach	Security parameter	Network/dataset	Experimental setup	Tools/simulators	Parameters/approach for intrusion detection	Limitations
DSSnet: microgrid simulator [1]	DoS and resilience	IEEE-13 bus power distribution system with two subsystems, i.e., wind turbine and energy storage system	Developed a simulator for evaluation of microgrid operation. Applications: (i) Self-healing network management (ii) Communication network verification (iii) Specification-based intrusion detection	OpenDSS, Mininet, virtual time system (Linux-based kernel)	(i) Network slicing (ii) Traffic isolation	A little literature on specification-based intrusion detection provided experimental validation of intrusion detection is not provided.
PYGRID: SG simulator [12]	DoS protection and resilience	Simulated IEEE-14 bus power system	Scenarios: normal operation, bus failure, and bus attack Result: successfully mitigated DDoS attack	Mininet, PYPOWER	(i) Number of packets/second = 40% threshold (ii) Flows count	(i) Maximum power capacity allowed on each bus/branch is not mentioned; rationale for using fixed threshold limit for number of packets/sec is missing. (ii) All traffic flows are being monitored for rapid detection. Computation overhead cost is associated with the approach since all flows will go to application layer. (i) Experimental validation for backup controller functionality, in case of primary controller failure, was missing. (ii) Threshold value should have been changed dynamically as per the changing network environment. (iii) The authors did not address the efficacy of approach in protecting against LR-DDoS attacks (iv) Flash crowds may be detected by the algorithm as an attack, resulting in extra FPR. (v) The proposed approach should have been validated against performance metrics like DR, FPR, etc.
Multicontroller-based SDN [20]	UDP/TCP/ICMP flood attacks	Simulated	Design components entropy-based DDoS detection algorithm (i) Virtualized network environment of 3 switches and 32 hosts (ii) Set of mitigation actions (block traffic/ports) (iii) UDP flood attack simulated	POX controller, Mininet 2.0, and Scapy tool for traffic generation	Analysis metric (i) Destination IP address entropy	

TABLE 2: Entropy-based approaches.

Year	Technique	Anomalies addressed	Dataset	Data source	Source tool	Flow properties for anomaly detection	Comparison	Validation metrics (%)	Conclusion
2016 [21]	Generalized entropy	DDoS, probe	Real: KDDcup99, NSL-KDD, UCI machine learning repository datasets Simulated: Testbed dataset (TUIDS) for DDoS and probe attacks	IP packet/ IP flow	Netflow data	Dynamic selection of features through mutual information and GE	LOF for $\tau = 0.58$ at dataset Zoo ORCA Proposed approach Shannon entropy Kullback–Leibler divergence	DR = 82.35 FPR = 19.04 DR = 88.23 FPR = 13.09 DR = 94.11 FPR = 2.38 DR = 55 FPR = 15 DR = 70 FPR = 15	Proposed approach achieved better DR and FPR metrics compared to other outlier approaches
2015 [22]	Extended entropy	DDoS, port scan, network scan, DoS, worm, and spam	Legitimate traffic from tsinghua University Campus network	IP flow	Netflow	Source IP address, source port, destination IP, address, destination port, flow byte, flow direction, protocol number, and TCP control bit	—	DR = 93.46 FPR = 5	2015
2017 [23]	Tsallis entropy	Real and simulated versions: DDoS, alpha flow, port scan, network scan	Real Campus network data, i.e., UTFPR/ Toleda Campus and FISTSC/GW campus	IP flow	Netflow v9	Source address, destination address, source port, destination port, number of packets, number of flows, number of bytes, in-degree	Tsallis entropy Shannon entropy After incorporating sampling effects in technique	DR = 100 FPR = 1 DR = 25 FPR = 2.2806 DR = 99.45 FPR = 0.12	Achieved better DR and FPR compared to Shannon entropy validation metrics dropped a little with sampling effects

SG has grabbed more intention of researchers. The reason is the occurrence of massive DDoS attack on Ukraine power grid in 2015. Existing security protocols/techniques provide network protection at Internet edge only and are not sufficient enough to prevent dynamic attacks, considering borderless architecture of IoT. Additionally, current approaches of security, i.e., firewall zoning and intrusion detection and prevention system (IDPS) are too constrained by traditional network architecture. They are computationally heavy when considering the increase in network devices [27]. If appropriate security actions are not taken, then attacks like DDOS, service unavailability, and most importantly threat to human life might happen. Moreover, early detection and mitigation are deemed necessary for infrastructure like SG since deep penetration to SG network can lead to devastating consequences.

Entropy-based techniques used by the researchers rely on experimental-based thresholds and do not adapt to

changing network conditions. Moreover, utilization of static experimental thresholds and Shannon entropy do not provide adequate security against both HR and LR-DDoS attacks for an ICT infrastructure like SG. Static thresholds need to be reconfigured on changing network conditions to adjust for high DR and low FPR and that makes it unsuitable for SG. Moreover, Shannon entropy provides low DR and FPR as compared to Tsallis entropy [23] and on detection of DDoS attack, it is important to mitigate it as well to prevent its penetration further in the network; that is missing in DDoS protection approaches. In order to improve the security and reliability of SG in reference to DDoS attacks, researchers have suggested an SDN-based approach to handle the glitches in the conventional network paradigms. However, these approaches [6, 12, 20] are still only capable of handling HR-DDoS attacks, i.e., TCP/UDP/ICMP-based flooding attacks only, not catering stealthy and low-rate DDoS attacks, and also rely heavily on static thresholds.

Hence, it makes it necessary to develop a light-weight DDoS defence mechanism for SG that is fueled by SDN environment and using Tsallis entropy for better DR and FPR. Additionally, the SDN environment should be adaptive and capable of providing early detection and mitigation of both HR and LR-DDoS attacks.

2.2. Solution and Contributions. Considering our proposed solution, DDoS detection application uses Tsallis entropy metric with traffic features, i.e., Source Address (SA), Destination Address (DA), Source Port (SP), and Destination Port (DP) for efficient detection of varying DDoS attacks. For mitigation approach, IP address and port blocking mechanism is available in SDN controller software; i.e., OpenFlow is utilized. Blocking data is provided by the local list maintained in the SDN controller. Since SDN controller provides a global view of the whole network and is centrally located, the proposed approach significantly addresses the locality problem of DDoS defense mechanism that is missing in literature. A novelty in approach is added by using dynamic thresholds for traffic features using Exponential Weighted Moving Average (EWMA) instead of static threshold values for detection purposes. Moreover, to the best of our knowledge, Tsallis entropy in SDN environment has not been used previously. The proposed approach provides a near real-time detection within 250 packets with mitigation of anomaly in real time. In the following section, the proposed system model is discussed in detail. The following contributions are made in this work:

- (i) A light-weight entropy-based detection approach is developed underlying SDN environment
- (ii) Adaptive threshold mechanism is proposed to achieve better DR and False Positive Rate (FPR) using Exponentially Weighted Moving Average (EWMA) and Tsallis entropy
- (iii) Low-rate (LR)- and HR-DDoS attacks are successfully detected
- (iv) In addition to the real-time protection mechanism, a DDoS mitigation mechanism is also explained in terms of proposed model
- (v) Resource utilization (CPU and RAM utilization is optimized without compromising LR- or HR-DDoS protection)

3. System Model

In existing SDN-based solutions, a limited level of DDoS protection, i.e., against flooding-based DDoS attacks only, is being provided. Further, deployment locality of DDoS defence mechanism is critical in efficient and in-time detection. Most of the researchers did not fully address the issue of locality. SDN controller has a global view of the network and is responsible for all routing and filtering features of a network. In other words, it is a brain and central point of network. Therefore, SDN network utilization can provide optimal deployment locality for DDoS defence mechanism.

Lastly, software-based control of SDN provides IP address/port blocking mechanisms as a built-in feature.

So, these mechanisms can be optimally utilized as a DDoS mitigation approach. The conceptual framework is divided into two parts, namely, SDN-based environment for consumer-utility provider network and intrusion detection and prevention system (IDPS) as depicted in Figure 1. In this section, a detailed description of the system model is presented, following the implementation constraints and limitations.

IDPS is divided into three modules, namely, flow collector ("FC"), anomaly detector ("AD"), and anomaly mitigation ("AM"), as depicted in Figure 1. "FC" module is located in controller and collects network flows/packets and statistics from each connected switch through Netflow standard protocol, utilized by the controller. These flows are stored in the local database of the controller and relevant features, i.e., Source Address ("SA") and Destination Address ("DA"), are extracted for further processing by "AD." "AD" calculates Tsallis entropy value per traffic feature in current window of 50 packets and compares it with corresponding feature threshold value for that window. In case of a mismatch as per conditions discussed in subsequent section, an alarm is generated and further action is taken by the "AM" module. "AM" module performs drop/deny action on the flows and pass it on to v-switch performing forwarding decisions. It also stores the blacklisted IPs in blacklist database maintained locally in the controller for scrutiny of incoming packets. Threshold calculator calculates threshold values per feature for next window by applying Exponentially Weighted Moving Average (EWMA) on current window entropy value and previous window threshold value and passes it on to "AD" module for comparison purposes. Table 3 describes the primitive flow properties being used in the analysis. After extraction of required details, data is parsed to "AD" module which follows the mechanism as discussed in upcoming sections.

3.1. Anomaly Detector Module (AD Module). After extraction of traffic features ("SA," "SP," "DA," and "DP") by "FC" from new packets destined to the controller from OF switches through Netflow protocol supported by POX controller, data is fed to the "AD" module. Data in anomaly detector is processed based upon window size that can be based upon either time stamp of packet received or number of packets. For this work, it is based upon number of packets and set to 50 packets per window for efficient detection and memory foot-print [20]. Moreover, the experimental setup constitutes not more than 50 hosts (smart meters and utility server), so 50 packets per window is an appropriate window size. Therefore, consider W as the set of data with n elements in which each data element x_{mi} signifies the event pertaining to specific traffic feature as can be seen in (1). Probability of x_{mi} happening in window W can be calculated using (2). Further, Tsallis entropy is denoted by " H_q " which can be calculated by (3) [23]. For $q > 1$, higher probabilities have more impact on the final entropy value compared to lower

FIGURE 1: System model S-DPS.

TABLE 3: Primitive flow properties for AD.

No. (m)	Primitive flow property (xm)	Detail
1	SA	Source IP address of packet
2	DA	Destination IP address of packet

probabilities and vice versa. Here value of q is set as -1.3 or -0.8 for high DR and low FPR.

$$w = x_{mi}, \quad m = 1, 2, 3, 4, i = 1, 2, \dots, n, \quad (1)$$

$$P_{mi} = \frac{x_{mi}}{n}, \quad (2)$$

$$H_q = \frac{1}{q-1} \left(1 - \sum_{i=1}^n p_{mi}^q \right) \quad (3)$$

In each window, entropy values of four traffic features are calculated and compared with respective normal entropy value, using (4). Here $H_{q-(n)}^{x_m}$ is the entropy value of specific traffic feature taken in normal traffic conditions, i.e., without any abnormal traffic, and λ signifies the difference of entropies. In case the value of λ is positive, it means the entropy value of feature for current window has decreased; i.e., data distribution is concentrated. However, in case value of λ is negative, it means the

entropy value of feature for current window has increased; i.e., data distribution is dispersed.

$$H_{q-(n)}^{x_m} - H_q = \lambda. \quad (4)$$

Application of (4) is different for each traffic feature as depicted in Table 4. In case of DDoS attack, value of λ is positive for “DA” and negative for “SA,” whereas for different types of DDoS attacks, values of “SP” and “DP” are variable. Equations (1)–(4) are calculated for subsequent windows (50 packets per window) and in case value of λ is positive for “DA” and negative for “SA” for five consecutive windows, an alert for DDoS attack is generated. A counter for subject purpose is utilized, which is incremented on meeting the set conditions in each window. In case set conditions for “SA” and “DA” are not met in any 5 consecutive windows, counter is set to zero and process starts again with counter = 0.

3.2. Anomaly Mitigation Module. A specific action is associated by the controller with each flow in flow tables of the controller as discussed in background section related to OF protocol. In case an alarm is generated by the “AD” module, then the “SA” with maximum number of occurrences in the 5 windows is extracted and “drop/deny” action is set by the controller against the matched flows associated with that “SA” in run time.

TABLE 4: Interpretation of value of λ .

No. (m)	Flow property	Value of λ	Impact	Result
1	SA	Negative real number	Data dispersion	Attack from multiple IPs
2	DA	Positive real number	Data concentration	Attack towards specific IP

3.3. Dynamic Threshold. Initially, threshold values for each traffic feature are set by simulating the network environment in normal conditions, i.e., without any attack traffic. These threshold values are used to detect DDoS attack in progress as per the conditions discussed previously. Value of threshold dictates the performance of entropy-based detection approach in terms of DR and FPR. So, choosing optimal thresholds is most significant and important to achieve desired results. One approach is to conduct multiple experiments using attack traffic (tool or datasets) with normal traffic to tune these thresholds, while another approach is to utilize current network conditions in real time and system automatically updates these thresholds. The latter is more convenient and effective, considering the dynamic nature of SG network. So, in order to make the anomaly detection adaptive, consider a mean entropy value for each traffic feature as and for each subsequent window, mean entropy value for each traffic feature as a threshold, is calculated using (5). EWMA filter is used for calculating the average mean, and β value of 0.1 is used for catering current network conditions and is more reactive in nature, considering highly critical networks such as SG. Value of constant c depends upon the network characteristics.

$$H_{q(i)}^{x_m} = \left(\beta \times H_{q(i-1)}^{x_m} + (1 - \beta) \times H_{q(i)}^{x_m} \right) + c. \quad (5)$$

Threshold values, calculated as per (5), are based upon current network conditions with β value set to 0.1 (very reactive) and can result in high FPR for burst channel. Similarly, in case of stealthy attack pattern such as increasing and decreasing DDoS attacks, detection will be difficult. So, there is a need to tune the value of threshold in real time. In order to achieve optimum DR and FPR and keep the threshold in acceptable bounds, a maximum change/difference of current threshold from the normal entropy threshold (calculated during normal conditions) should not exceed by 1.5, considering 90% confidence interval for normal distribution. In case it exceeds more than 1.5 times, value of current threshold will become 1.5 times to normal entropy threshold; otherwise, it will remain the same as per the calculated mean threshold value. However, for decreasing entropy with respect to normal entropy threshold (calculated during normal conditions) for more than 1.5 times, value of current threshold will be normal entropy threshold value divided by 1.5 to normalize the threshold; otherwise, it will remain the same as per the calculated mean threshold value. The multiplication and division factor of 1.5 is used to keep the thresholds within reasonable bounds with respect to normal threshold value. The increasing entropy check is applicable for SA entropy, whereas decreasing entropy check is applicable for DA entropy. The reason is that DDoS attack tends to decrease DA entropy while

increasing SA entropy values. Flow chart for the algorithm is presented in Figure 2. In OF-based v-Switch, a packet for which no flow entry exists is passed on to controller for decision making. So, in the algorithm packet in flow step signifies entry of new packet in the controller. Traffic features of the packet as per Table 3 are checked for existence of entries already in the system. In case entries exist in the lists; then occurrence counters for each feature are incremented. Otherwise, new entries in the corresponding lists are made. If the number of packets count has reached 50 as per the set window, entropy value for each traffic feature using the corresponding traffic feature list is calculated. It is then compared with the threshold value. In case current DA entropy value is less than mean DA threshold value and current SA entropy value is greater than mean SA threshold value; then the consecutive window counter is incremented and the same cycle starts again for next window with number of packets count set to zero. Moreover, in case the current DA, entropy value is greater than mean DA threshold value and SA entropy is less than mean SA threshold value; then the cycle starts again with number of packets count set to zero.

3.4. Implementation Constraints and Limitations. As previously mentioned in related work, DDoS related datasets for SG are not publicly available and datasets like MIT Lincoln, FIFA, DDoSTB, and CAIDA datasets are not SG related [12]. Therefore, [1, 12] relied on simulated traffic to test the viability of their proposed approach. Similarly, in the paper, simulated normal and attack traffic is being generated using Scapy tool to test the proposed model because it is python-based and can be integrated with Mininet. Single topology is tested for different types of DDoS attacks and the traffic is simulated one. Results obtained may vary in real-time traffic. Moreover, model presented is independent of any protocol (tested for TCP/UDP/ICMP-based packets) and threshold for DDoS detection is being adjusted automatically with varying network conditions. So, solution is viable for dynamic network conditions as in modern networks. Apart from it, the solution is tested using a software-based simulator. Its capability will further be improved with powerful hardware-based SDN controller available in the markets.

Furthermore, the approach is based on single controller architecture, wherein it can present a bottleneck and security constraint when dealing with large-scale network like SG. Difference between using single-controller and multicontroller architecture is linked to load balancing, high availability, and security of controller. However, for this research it is outside the scope of work and the approach can be integrated and tested with multicontroller architecture for future research.

FIGURE 2: Proposed IDS: flow chart.

4. Experimental Setup

In this section, an experimental setup for validation of S-DPS against Utility-Consumer Communication Network implementation is discussed. For this purpose, a series of steps are followed in order to establish simulation for performing the experiments using test cases as discussed in the following section.

4.1. Simulation Steps

4.1.1. Controller. POX is used as SDN controller for the experiments. It is an open-source and python-based controller that is widely used in experiments. It is lightweight and developed as a platform to be customizable, meeting desired needs of a controller. It supports famous operating system like Windows, Linux, and MAC OS and has a network discovery feature installed. Apart from this, another two famous controllers like Floodlight and Beacon are also available. However, in most SDN-based papers highlighted in literature review, NOX controller, a predecessor of POX, is used. So, for the research POX controller is selected.

4.1.2. Network Emulator. Mininet 2.2.2 is used as a network emulator for the experiment. It is an open-source platform with support for SDN environment and OF protocol. It treats each network component as a kernel process and can be installed easily on a laptop or Personal Computer (PC) using kernel namespace feature. Each network namespace has its own Network Interface Card (NIC), Address Resolution Protocol (ARP) table, ping service, scripts, and routing table. Both Graphical User Interface (GUI) and command line interfaces are available to create network topologies. As a default, NOX controller is embedded in Mininet.

4.1.3. Traffic Generation. Scapy is used as a traffic generator tool, both for normal and for attack traffic. It has features of scanning, packet spoofing, packet forging, sniffing, etc. Here, TCP packets are generated using the tool. It supports python programming language and POX controller also uses python. So, both controller and traffic generation tool can be integrated. Spoofed source IP addresses and Host IP addresses are generated using python function "random."

This function returns a uniform random float in the range of 0.0 to 1.0. These random floats are joined together to form a spoofed IP address. Other options, i.e., type of packets and packets interval available in Scapy, are used to create normal and attack traffics. TCP/UDP/ICMP is set for type of packets and 0.4 seconds as interval for normal traffic between smart meter and utility server. Moreover, TCP/UDP/ICMP-based DDoS attacks with attack rates ranging between 200 and 4000 packets/sec are simulated in existing researches, i.e., [6, 20, 27, 28]. Such variations of traffic generation are catered for in existing experiments, covering both LR- and HR-DDoS attacks.

4.1.4. Network Setup. Network is set up on Laptop Dell Inspiron with Core i3 2.41 Ghz processor, 4 GB RAM, and 100/1000 Gbps NIC. Operating System is Windows 8.1 with VirtualBox 6.0.4 installed. Mininet 2.2.2 on Linux Ubuntu 14.04.4 is installed in the VirtualBox for setting up the environment. Mininet 2.2.2 supports OF version 1.3. Moreover, "mn" command is used in Mininet to set up the network. As a default, two hosts with one switch are configured. However, custom network is set up using different filters available in "mn" command, i.e., related to controller either local or remote, type of switch, number of hosts/switches, etc.

4.1.5. Network Topology. A tree-type network constituting smart-meter-utility server communication network is depicted in Figure 3. It has a depth of 2 with 10 switches and 54 hosts (smart meters and utility server). Here "smart meters" are the core of SGs because they are smart and possess the ability to sense, measure, and examine the usage of electricity, continuously transmit the data and information collected to the central location, and perform two-way communications with all other components of the SG and the consumer. Meanwhile, "utility server" has a dual-role to play; i.e., it has a two-way communication with smart meter as well as with power generation facility. It is located at control center and provides live consumption data to both users and to power generation facility. Finally, "controller" is the brain of the overall network managing all OF-enabled switches/routers by installing forwarding rules and performs centralized network and configuration management for better performance and security in the network.

Utility server is connected to Switch-1, whereas Switches 2–10 are used to connect 53 smart meters, evenly divided, i.e., 6 smart meters each. However, last switch consists of 5 smart meters. OF-enabled v-Switch available in Mininet is used to connect hosts. L3-learning module of POX controller with addition of two functions, i.e., traffic feature collection and entropy calculation, is used for the controller function of the network.

4.2. Evaluation Criteria. The S-DPS is evaluated using DR and FPR metrics where DR measures portion of the attacks that are detected correctly by the system and represented by (6) and FPR provides the percentage of events that are reported as negative events where actually they are positive events and represented by (7). In the equations, True Positive (TP) event means that the system has detected a correct anomalous event, whereas False Positive (FP) means system has detected an incorrect anomalous event; i.e., actually the event is legitimate but detected otherwise. Similarly, True Negative (TN) event means that the system has detected a correct legitimate event, whereas False Negative (FN) means system has detected an incorrect legitimate event, i.e., actually the event is anomalous but detected otherwise. Varied levels of both LR- and HR-DDoS attacks, i.e., smurf, socket stress, and SYN flood attacks, are launched against the utility server for early detection and real-time mitigation of attack.

FIGURE 3: Network topology.

$$DR = \frac{TP}{TP + FN}, \quad (6)$$

$$FPR = \frac{FP}{TN + FP}. \quad (7)$$

5. Results and Discussion

The experiment covers topology highlighted in Figure 3, which contains 10 switches and 54 hosts. Each host from h1-h53 represents a smart meter, where host h54 is a utility server with which each smart meter sends its observed values. Each switch in the topology is OF-enabled v-Switch centrally connected to POX controller c0. In order to simulate the traffic between connecting entities, certain parameters like frequency of communication between smart meter and utility server, type of protocol, etc., need to be defined. AMI infrastructure does not have any standardized architecture and varying implementations exist defining the network and dynamics of communication. Frequency of communication between smart meter and utility server is also set to different intervals, i.e., 1 second, 4 seconds, 60 seconds, 5 minutes, and 15 minutes, depending upon the scheduling criteria set by utility service provider [29]. Considering the periodic traffic profile in most common architectures, smart meters are scheduled to transmit and receive at interval of 0.4 seconds [30]. Further, both UDP and TCP protocols are used in two-way communication between smart meter and utility server. Seven sets of traffic profiles are generated in the experiment, i.e., normal traffic, smurf attack, socket stress attack, and SYN flood attack. Traffic profile for the experiments is shown in Table 5. These traffic profiles are

simulated using UDP/TCP/ICMP-based packets at destination port 80/21 using random spoofed source IP addresses and source ports.

5.1. Normal Traffic Profile. In normal traffic profile, a total of 5 runs of experiment are performed, each containing 1250 packets with window size of 50 packets. Packet interval between smart meter(s) and utility server and reverse is set to 0.4 seconds. In normal circumstances, at any given point in time, a utility server is sending probe to any smart meter and any smart meter is sending its readings to utility server.

Therefore, two-way packets are generated randomly using one of the IP addresses of smart meter and of utility server with interval of 0.4 seconds to obtain average normal entropy value. The whole experiment is covering observation of 6,250 packets. The results of normal traffic separately for source IP (SrcIP) and destination IP (DestIP) are presented in Figure 4. Here, average entropy values per window for both SrcIP and DestIP are used to plot the graphs. As can be seen from Figure 4, entropy value for DestIP ranges from 1011.923 to 1372.990 and average normal entropy value is being utilized as a base entropy for DestIP in attack scenarios. Similarly, entropy value for SrcIP ranges from 1066.081 to 1722.328 and average normal entropy value is 1320.678, being utilized as a base entropy for SrcIP in attack scenarios.

5.2. Smurf Attack. A smurf attack is a type of DDos attack in which vulnerabilities in Internal Protocol (IP) or Internal Control Message Protocols (ICMP) are exploited as such that it makes the overall computer network inoperable. For

TABLE 5: Traffic profiles.

Type of traffic	Protocol	DP	SP	Payload: number of packets	Source IP address	Destination IP address	Traffic interval	Attack type
Normal	UDP	80/21	2/3	None	10.0.0.54 or random (10.0.0.0/24)	10.0.0.54 or random (10.0.0.0/24)	0.4 sec	—
Smurf attack	ICMP	—	—	6000 bytes	10.0.0.54	10.0.0.255	—	DDoS
Socket stress	TCP	21	Random (0–65535)	None	10.0.0.4	10.0.0.54	—	DDoS
SYN flood	TCP	80	Random (1000–9000)	None	Random	10.0.0.54	—	DDoS

FIGURE 4: Normal traffic profile.

smurf attack to work, a false IP packet with spoofed IP is created. IP packet is basically an ICMP ping message that tells the network nodes to receive and send back echo reply. These echoes are then sent back to all network devices creating an infinite loop in the network. To further amplify the attack, IP broadcasting technique can be used.

In the experiment, an ICMP echo request is generated towards the broadcast address of all switches/routers, i.e., 10.0.0.255 using the spoofed IP address, i.e., of target address (10.0.0.54), which is a utility server. In this case, all the smart meters lying under these switches/routers will send their ICMP echo replies towards the target, i.e., utility server. In order to further amplify the attack, each smart meter is relaying 6000 bytes of junk IPv4 packets towards the target. Two separate scripts are being run manually using two random hosts, e.g., h1 and h4. At h1, normal traffic generation is carried out, whereas at h4 (attacker machine), smurf attack towards target address (utility server) is launched. Traffic profiles both for source and destination IP for the scenario are depicted in Figure 5. It can be seen from Figure 5(b) that destination IP current entropy is far below the threshold value between windows 6 and 25, meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in

these windows. Further, to verify whether it is a DoS or DDoS attack it can be seen from Figure 5(a) that source IP current entropy is above the threshold value between windows 11–22, meaning the number of packets with multiple SrcIPs/windows for the target host exists, resulting in increase of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DDoS. In case it is below the threshold values, the attack is considered as DoS attack.

Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 5(c). It can be seen that entropy values/window for attack traffic has declined significantly compared to normal traffic since most of the traffic/window is directed towards a single DestIP, resulting in decline of entropy. Moreover, it can be observed from Figures 5(a) and 5(b) that S-DPS-based threshold is changing as per the current network conditions compared to experimental static threshold that remains fixed no matter how much the network environment varies. So, S-DPS-based threshold is able to provide true picture of the network while achieving DR of 100% with 0% FPR for simulated traffic.

5.3. Socket Stress Attack. Considering socket stress attack, raw sockets are used to establish a connection with the target machine. It is an asymmetric resource consumption attack,

FIGURE 5: Smurf attack detection- static vs. dynamic thresholds. (a) Source IP address traffic profile; (b) destination IP address traffic profile; (c) destination IP-normal vs. smurf attack traffic.

where asymmetric refers to less requirement of resources at attacker end versus a great deal of resource consumption on target machine. For such attack to work, it should be targeted to an open port in victim's machine. In the attack, attacker advertises a zero window at the end of three-way handshake; meaning it has not received the data so the victim will tend to open the connection and probe the client periodically to check whether data is received or not.

Similarly, multiple connections at the victim machine are opened, consuming many resources on the victims' machine. Socket stress attack script is executed randomly on h4 (*attacker machine*) targeting utility server (*victim machine*) at IP address 10.0.0.54 and port 80. In the attack, 20 random connections using random source ports are created with a timeout value of 1 minute. Timeout value defines the time before which new connection is established to the target. So,

at any given point in time, a minimum of 20 connections remain active on the target machine. Two separate scripts are being run manually using two random hosts, say h1 and h4. At h1, normal traffic generation is carried out, whereas at h4, socket stress attack towards target address is launched. Traffic profiles both for source and destination IP for the scenario are depicted in Figure 6. It can be seen from the figure that destination IP current entropy is far below the threshold value between windows 5 and 25, meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in these windows. Further, to verify whether it is a DoS or DDoS attack, it can be seen from Figure 6(b) that source IP current entropy is not above the threshold value for consecutive windows from windows 1 to 25. It means that the number of packets with single SrcIP/window for the target host exists, resulting in decrease of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DoS. Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 6(c). It can be seen that destination IP entropy values/window for attack traffic has decreased significantly after the attack compared to normal traffic using the proposed S-DPS mechanism since most of the traffic/window is directed towards a single DestIP, resulting in decline of entropy.

5.4. SYN Flood Attack. In case of SYN flood attack, the attacker exploits part of normal TCP three-way handshake process by sending repeated SYN packets to the target machine with a frequency above its capacity to process. It can target all open ports or a specific port to block the service(s) of the target machine. The target machine responds to all received requests with SYN-ACK packets for that open port(s) and wait for ACK packets for some time. In most scenarios, source IP address and ports are malicious, i.e., spoofed, so ACK packets are never sent back or, in another case, ACK packets are not sent by the attacker deliberately to shut down the service of target machine. SYN flood attack script is executed randomly on h4 (*attacker machine*) targeting utility server (*victim machine*) at IP address 10.0.0.54 and port 80. In the attack, 10,000 packets with random source IP address and ports (ranging between 1000 and 9000) are sent to the utility server. These packets have random “seq” numbers and “window” size between 1000 and 9000. Two separate scripts are being run manually using two random hosts, say h1 and h4. At h1, normal traffic generation is carried out, whereas at h4 (*attacker machine*), SYN flood attack towards target address (*utility server*) is launched. Traffic profiles both for destination and source IPs for the scenario are depicted in Figure 7. It can be seen from Figure 7(a) that destination IP current entropy is far below the threshold value between windows 5 and 25; meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in these windows. Further, to verify whether it is a DoS or DDoS attack, it can be seen from Figure 7(b) that

source IP current entropy is above the threshold value for consecutive windows from windows 6–25, meaning the number of packets with multiple SrcIPs/windows for the target host exists, resulting in increase of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DDoS. Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 7(c). It can be seen that destination IP entropy values/window for attack traffic has decreased significantly after the attack compared to normal traffic using the proposed S-DPS mechanism since most of the traffic/window is directed towards a single DestIP, resulting in decline of DestIP entropy.

For all the attacks discussed above, although the target is *utility server (h54)*, controller being the brain of SDN network is processing all the normal and attack packets. So, in addition to utility server (*target machine*) controller is also being targeted in all attack scenarios discussed, but the detection and mitigation approach is implemented at the controller so attack is mitigated within near real time.

5.5. Mitigation of DDoS Attack. On detection of DDoS attack, it is important to mitigate it as well to prevent its penetration further in the network. OF protocol, due to its real-time reconfiguration feature, enables us to define flow rules that can block the switch ports in real time. The authors in [6, 27, 28] utilized OF port blocking or deletion of flows as a mitigation strategy, achieving time and space complexity of $O(n)$, where “ n ” may be number of packets processed for port blocking or number of flows deleted. For that matter, port blocking strategy is utilized, achieving the same complexity of $O(n)$. One timer variable of Boolean type, i.e., “timerSet,” and two functions, i.e., Preventing() and _timerfunc(), are incorporated in the default L3_learningmodule of POX controller. By default, timerSet is set to “False” so that controller continues to operate without active DDoS defense mechanism till entropy of the window does not fall under threshold value of that window. In case entropy values of the windows from the entropy dictionary are less than the threshold values, Preventing() function is invoked with global Set_Timer set to “True”; otherwise, timerSet value is set to “False” to enable/allow normal operation of the controller again, i.e., without active DDoS defense mechanism. Eventually, _timer_func is used to detect the happening of DDoS attack using the dictionary maintained by Preventing() function and block the switch ports with count greater than and equal to 5, occurring in five consecutive windows. Preventing() function is incorporated in POX controller using “_handle_openflow_packetIn” instance. Each time a new packet enters the controller, packet is accounted for in the dictionary being maintained. Dictionary constitutes a switch ID and port number with its counter. It has a form like switch ID (port number, count). Switch ID is recognized by OF parameter “event.connection.dpid” and port number by “event.port.” It is used to detect whether DDoS attack has occurred or not. After creating the dictionary for 25 windows, _timer_func() is used to detect and mitigate

FIGURE 6: Socket stress attack detection-static vs. dynamic thresholds. (a) Destination IP address traffic profile. (b) Source IP address traffic profile. (c) Destination IP-normal vs. socket stress attack traffic.

DDoS attack. It iterates through all the items in the dictionary and if specific ports of a specific switch have its count greater than and equal to 5 and for five consecutive windows, DDoS attack is detected and these switch ports are blocked by sending message to controller using OF

procedure calls, i.e., “*of.of p_packet_out*” and “*core.open_flow.sendToDPID()*.” The mitigation strategy is performed successfully on 25% rate attack on single host. As per results, dictionary maintained by the controller contains count of 69 for Switch-1 and Port-1, 12 for Switch-2 and

FIGURE 7: SYN flood attack detection-static vs. dynamic thresholds. (a) Destination IP address traffic profile; (b) source IP address traffic profile; (c) destination IP-normal vs. SYN flood attack traffic.

Port-1, 60 for Switch-2 and Port-4, and 61 for Switch-8 and Port-9. All such switch ports are blocked by the controller as part of prevention strategy.

6. Performance Evaluation

The performance of S-DPS is evaluated using metrics like CPU/RAM utilization. CPU/RAM utilization is measured

and compared with and without the approach using 25% attack rate on single host scenario. As mentioned in previous sections of conceptual framework, 4 additional functions are added to the L3-learning module of POX controller, i.e., traffic feature collection, entropy calculation, timer function, and preventing function, for DDoS detection and mitigation purpose. In order to see the effect of these functions on the overall CPU/RAM utilization of Mininet and on the

controller, two simulations are run again. One simulation constitutes 25% rate attack on single host without the solution and other simulation with same setting with proposed solution. The elapsed time for both simulations is 25 seconds and normal traffic ran for 200 seconds. “Top” and “Htop” commands have been used to capture the CPU/RAM utilizations. Results are depicted in Table 6. It can be seen from Table 6 that overall CPU/RAM utilization is 55.5%/171 MB in case of simulation without the solution and controller instance has consumed 12.3%/1.4% of total memory. In case of simulation with the solution, overall CPU/RAM utilization is 55.2%/205 MB and controller instance has consumed 29.6%/1.7% of total memory. There is a slight increase in controller instance CPU/RAM utilization but it is still in acceptable limits.

DDoS detection and mitigation functions are incorporated in SDN controller, considering the low computational complexity of approach used, i.e., $O(n)$ for both time and space complexity. It is verified by CPU/RAM utilization with and without the approach. At controller end, CPU utilization rises to only 29.6% from 12.3% with S-DPS. Similarly, there is a minimal increase in RAM utilization from 1.4% to 1.7%. Considering the facts, S-DPS can be both efficient and effective approach to provide DDoS protection in dynamic networks like SG communication network. The reason is its nondependency on any training requirements and due to adaptive nature of threshold calculations.

Several approaches to DDoS detection exist in literature. For example, Self-Organizing Maps (SOM), a machine learning approach, has been used by [31] to learn the behavior of network and decide whether network is attacked or not. Several hours of learning is required for better DR and FPR. In case of network or topology change, SOM is required to be trained again. With expansion of network, neurons used in SOM are also required to be increased, making the solution more expansive towards the network. The S-DPS is built in inside the controller and is easily adaptable to the changing network. No training is required upfront and computational complexity is lower than machine learning approach—SOM. Similarly, the authors in [32, 33] have utilized SNORT alongside SDN for DDoS detection. As highlighted previously, S-DPS has achieved better CPU/RAM utilization compared to SNORT. Moreover, DDoS protection mechanism is embedded in S-DPS, where in [33] separate SNORT detection system is integrated with SDN environment making it less transparent towards computational overhead, sampling requirements, and bandwidth limitations, if any. Both SOM and SNORT apply complex operations to learn the behavior of the network, e.g., processing large matrices or pattern matching schemes. In S-DPS, entropy-based mechanism is providing the same functionality without any of the complexities available in SOM and SNORT.

Benefits that are achieved through S-DPS are highlighted as follows:

- (i) High DR with no FPR
- (ii) DoS, LR-DDoS, and HR-DDoS attacks that have been successfully detected

TABLE 6: Resource utilization.

Resource utilization	Controller instance		Mininet instance	
	CPU%	RAM%	CPU%	RAM (Mbs)
Without S-DPS	12.3	1.4	55.5	171
With S-DPS	29.76	1.7	55.2	2.5

- (iii) Threshold mechanism that is adaptive rather than static and without any experimental adjustment for better DR/FPR, thus making it more suitable for modern/dynamic networks
- (iv) DDoS mitigation mechanism also provided as an addition for real-time protection

7. Conclusion

Given the nature of current dynamic networks, DDoS attacks are constantly becoming more sophisticated and are rapidly growing. These attacks can prove to be devastating for the underlying networks, with special emphasis to communication networks existing in SGs. The conventional approaches to counter these attacks are not enough to provide sufficient safety. Many researchers have claimed that the evolving SDN-based approaches are successful in dealing with the DDoS attacks in such networks. But, it is reported that these approaches employ the static threshold mechanisms to detect the attacks which is not suitable, given the dynamic and heterogeneous networks of SGs. S-DPS has claimed to efficiently address and manage these issues by employing an SDN-based environment and using a lightweight entropy-based defense mechanism. The DDoS protection strategy is made more efficient and effective through the reconfiguration of network in real time and by providing the global view of SDN networks. It is capable of detecting the threat along with the mitigation of anomaly at the same time as early as the first 250 packets by blocking the ports. Additionally, the existing SDN-based approaches are unable to detect different level of DDoS attacks but with the use of Tsallis entropy and its sensitivity factor, detection becomes possible. DR of 100% with FPR of 0% is achieved through simulation of HR-DDoS attacks. The S-DPS is able to show its capability and productiveness in both protection against DDoS and computational costs through minimum usage of CPU and RAM.

7.1. Future Works. Single controller architecture is utilized in S-DPS, making it vulnerable to computational/bandwidth bottlenecks for very large networks. In order to add resiliency in S-DPS, a multicontroller architecture is recommended. Intercontroller communication mechanism is necessary to provide synchronized operations of the protection system, with necessary recovery and failsafe mechanism.

Data Availability

The data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (NSFC) under grant no. 62072217.

References

- [1] D. Jin, Z. Li, C. Hannon et al., "Toward a cyber resilient and secure microgrid using software-defined networking," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2494–2504, 2017.
- [2] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: a comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46595–46620, 2019.
- [3] H. Maziku, S. Shetty, and D. M. Nicol, "Security risk assessment for SDN-enabled smart grids," *Computer Communications*, vol. 133, pp. 1–11, 2019.
- [4] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: threats and potential solutions," *Computer Networks*, vol. 169, Article ID 107094, 2020.
- [5] M. Abujubbeh, F. Al-Turjman, and M. Fahrioglu, "Software-defined wireless sensor networks in smart grids: an overview," *Sustainable Cities and Society*, vol. 51, p. 101754, 2019.
- [6] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and 600s Flow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 601, pp. 122–136, 2014.
- [7] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of Things: evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, Article ID 101728, 2019.
- [8] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "A systematic review of data protection and privacy preservation schemes for smart grid communications," *Sustainable Cities and Society*, vol. 38, pp. 806–835, 2018.
- [9] M. A. Ferrag, M. Babaghayou, and M. A. Yazici, "Cyber security for fog-based smart grid SCADA systems: solutions and challenges," *Journal of Information Security and Applications*, vol. 52, p. 102500, 2020.
- [10] S. K. Singh, Y. S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, Article ID 102252, 2020.
- [11] F. A. Khan, M. Asif, A. Ahmad, M. Alharbi, and H. Aljuaid, "Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development," *Sustainable Cities and Society*, vol. 55, Article ID 102018, 2020.
- [12] M. Samir, M. Azab, M. R. Rizk, and N. P. Y. G. R. I. D. Sadek, "A software development and assessment framework for grid-aware software defined networking," *International Journal of Network Management*, vol. 28, p. e2033, 2018.
- [13] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425–441, 2017.
- [14] O. Jung, P. Smith, J. Magin, and L. Reuter, "Anomaly detection in smart grids based on software defined networks," vol. 1, pp. 157–164, in *Proceedings of the 8th International Conference on Smart Cities and Green ICT Systems*, vol. 1, pp. 157–164, SMARTGREENS, Heraklion, Crete, May 2019.
- [15] W. Chen, S. Xiao, L. Liu, X. Jiang, and Z. Tang, "A DDoS attacks traceback scheme for SDN-based smart city," *Computers & Electrical Engineering*, vol. 81, p. 106503, 2020.
- [16] X. Min, Z. J. Jin, R. H. Yang, M. R. Heng, and C. M. Xin, "Traffic scheduling strategy of power communication network based on SDN," *Computer Engineering and Applications Journal*, vol. 9, pp. 49–60, 2020.
- [17] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.
- [18] S. Ali and Y. Li, "Learning multilevel auto-encoders for DDoS attack detection in smart grid network," *IEEE Access*, vol. 7, pp. 108647–108659, 2019.
- [19] Z. Ahmed, N. Afaqui, and O. Humayan, "Detection and prevention of DDoS attacks on software defined networks controllers for smart grid," *International Journal of Computer Applications*, vol. 975, p. 8887, 2019.
- [20] T. Pandikumar, F. Atkilt, and C. A. Hassen, "Early detection of DDoS attacks in a multi-controller based SDN," *International Journal of Engineering Science*, p. 13422, 2017.
- [21] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243–271, 2016.
- [22] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "E-LDAT: a lightweight system for DDoS flooding attack detection and IP traceback using extended entropy metric," *Security and Communication Networks*, vol. 9, no. 16, pp. 3251–3270, 2016.
- [23] A. A. Amaral, L. d. S. Mendes, B. B. Zarpelão, and M. L. P. Junior, "Deep IP flow inspection to detect beyond network anomalies," *Computer Communications*, vol. 98, pp. 80–96, 2017.
- [24] I. Aouini and L. B. Azzouz, "Smart grids cyber security issues and challenges," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 9, pp. 1255–1261, 2015.
- [25] R. A. Niazi and Y. Faheem, "A bayesian game-theoretic intrusion detection system for hypervisor-based software defined networks in smart grids," *IEEE Access*, vol. 7, pp. 88656–88672, 2019.
- [26] M. Ghanbari and W. Kinsner, "Detecting DDoS attacks using polyscale Analysis and deep learning," *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 14, no. 1, pp. 17–34, 2020.
- [27] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson, "Flow based security for IoT devices using an SDN gateway," in *Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 157–163, IEEE, Vienna, Austria, August 2016.
- [28] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," vol. 1, pp. 310–317, in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 310–317, IEEE, Helsinki, Finland, 2015.
- [29] K. Balachandran, R. L. Olsen, and J. M. Pedersen, "Bandwidth analysis of smart meter network infrastructure," in *Proceedings of the 16th International Conference on Advanced Communication Technology*, pp. 928–933, IEEE, Pyeong-Chang, South Korea, February 2014.

- [30] P. Cheng, L. Wang, B. Zhen, and S. Wang, "Feasibility study of applying LTE to smart grid," in *Proceedings of the 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, pp. 108–113, IEEE, Brussels, Belgium, October 2011.
- [31] E. Mota, A. Passito, and R. Braga, "Lightweight DDoS flooding attack detection using NOX/Openflow," in *Proceedings of the IEEE 35th Conference on Local Computer Networks*, IEEE, Denver, CO, USA, April 2010.
- [32] A. Mahajan, A. Gupta, and L. Sen Sharma, "Performance evaluation of different pattern matching algorithms of snort," *International Journal of Advanced Networking Applications*, vol. 10, no. 02, pp. 3776–3781, 2018.
- [33] D. Huang, L. Xu, C. Chung, and T. Xing, "SnortFlow: a openflow-based intrusion prevention system in cloud environment," in *Proceedings of the Second GENI Research and Educational Experiment Workshop*, Salt Lake, UT, USA, March 2013.