Hindawi
Security and Communication Networks
Volume 2023, Article ID 9807607, 1 page
https://doi.org/10.1155/2023/9807607

Retraction

Retracted: An Adaptive Authenticated Model for Big Data Stream SAVI in SDN-Based Data Center Networks

Security and Communication Networks

Received 26 December 2023; Accepted 26 December 2023; Published 29 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Q. Zhou, J. Yu, and D. Li, "An Adaptive Authenticated Model for Big Data Stream SAVI in SDN-Based Data Center Networks," *Security and Communication Networks*, vol. 2021, Article ID 5451820, 14 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5451820, 14 pages
https://doi.org/10.1155/2021/5451820

Research Article

An Adaptive Authenticated Model for Big Data Stream SAVI in SDN-Based Data Center Networks

Qizhao Zhou,[1] **Junqing Yu**,[2] **and Dong Li**[2]

[1]*School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*Center of Network and Computation, Huazhong University of Science and Technology, Wuhan 430074, China*

Correspondence should be addressed to Junqing Yu; yjqing@hust.edu.cn

Received 20 May 2021; Revised 27 August 2021; Accepted 28 August 2021; Published 21 September 2021

Academic Editor: Chi-Hua Chen

With the rapid development of data-driven and bandwidth-intensive applications in the Software Defined Networking (SDN) northbound interface, big data stream is dynamically generated with high growth rates in SDN-based data center networks. However, a significant issue faced in big data stream communication is how to verify its authenticity in an untrusted environment. The big data stream traffic has the characteristics of security sensitivity, data size randomness, and latency sensitivity, putting high strain on the SDN-based communication system during larger spoofing events in it. In addition, the SDN controller may be overloaded under big data stream verification conditions on account of the fast increase of bandwidth-intensive applications and quick response requirements. To solve these problems, we propose a two-phase adaptive authenticated model (TAAM) by introducing source address validation implementation- (SAVI-) based IP source address verification. The model realizes real-time data stream address validation and dynamically reduces the redundant verification process. A traffic adaptive SAVI that utilizes a robust localization method followed by the Sequential Probability Ratio Test (SPRT) has been proposed to ensure differentiated executions of the big data stream packets forwarding and the spoofing packets discarding. The TAAM model could filter out the unmatched packets with better packet forwarding efficiency and fundamental security characteristics. The experimental results demonstrate that spoofing attacks under big data streams can be directly mitigated by it. Compared with the latest methods, TAAM can achieve desirable network performance in terms of transmission quality, security guarantee, and response time. It drops 97% of the spoofing attack packets while consuming only 9% of the controller CPU utilization on average.

1. Introduction

The big data streams have the characteristics of being security-sensitive, having data size randomness, and being latency-sensitive [1]. Current data-driven and bandwidth-intensive applications in data center networks [2, 3] become increasingly complex. SDN simplifies the application management by utilizing various policy-based controls over SDN-enabled access layer switches. With the rapid development of applications in the northbound interface of SDN [4], the big data stream is dynamically generated with high growth rates in SDN-based data center networks. The SDN-based policies are enforced through flow tables, which is specified by various flow entries and match fields [5]. However, the cloud servers in SDN-based data center networks may be attacked and return incorrect flow table query results. Spoofed source addresses could be used to prevent tracking, attack flow tables, and circumvent security checks [6, 7]. In addition, the recent big data stream growth, which transfers huge quantities of data between thousands of servers [8, 9], makes it more complicated for the spoofed address verification. Thus, how to ensure the integrity of big data streams in SDN-based data center networks without affecting normal communications SDN-based networks has taken a crucial role when performing policy-based communications in the SDN-based data center networks [10].

Internet Engineering Task Force (IETF) SAVI group[1] has standardized the access layer source address validation mechanisms. They designed a binding-validation model to

prevent IP spoofing and policy confusing. Specially, the SDN controller maintains the source address binding table in the binding-validation model centrally [11]. With this feature, SAVI is widely adopted to validate big data streams in SDN-based data center networks. However, the big data stream traffic may put high strain on the SDN-based communication system during larger spoofing events. Such unique characteristics raise new drawbacks and room for improvement for the implementation of big data stream SAVI:

(1) The data stream size generated by the data-driven and bandwidth-intensive applications is unpredictable. It is significant to determine the authenticated flow size in the big data stream in case the SAVI devices work separately and statically. Otherwise, the authentication performance will be inferior under big data streams in SDN-based data center networks.

(2) Big data stream is security-sensitive. However, binding relationships in the existing binding-validation model are unable to be exchanged between devices dynamically. It is crucial to provide robust security verification service under spoofing attacks to maintain the security level provided by dynamic SAVI (D-SAVI) [12].

(3) Previous D-SAVI scheme is unable to provide differentiated executions of the normal packet forwarding and the spoofing packet discarding. The policy-based controls over SDN-enabled switches under big data streams delivered by those bandwidth-intensive applications will lead to the packet forwarding efficiency reduction seriously.

In brief, both of the aforementioned limitations might incur additional overhead, thereby degrading SAVI performance under big data streams in SDN-based data center networks. In this study, we propose an adaptive authenticated model for big data stream source address validation, which optimizes the SDN-based network communication performance while maintaining the source address security level. Our main contributions in this paper are summarized as follows:

(i) We proposed a controller-based model for big data stream SAVI management and then provided an architectural design of a security mechanism that permits attack detection and source address validation implementation under big data streams. Our proposed model eases the collaboration not only between the forwarding entities, but also between networks. Therefore, spoofing attacks under big data streams can be directly mitigated.

(ii) The SPRT-based model is controlling big data stream SAVI by changing the validation implementation according to the anomaly classification, which will be expected to effectively balance the flow table utilization and the SDN controller response efficiency. The results are almost predictable since the model has classified various candidates to reduce the redundant big data stream SAVI process compared with the latest methods.

(iii) Our proposed method can defend against the spoofing attack effectively and make it possible for the controller to mitigate ongoing attack actively. Experimental results show that it can identify the attack and legitimate traffic with high accuracy. It drops 97% of the spoofing attack packets while consuming only 9% of the controller CPU utilization on average.

The paper is structured as follows. Section 2 describes the adaptive authenticated model for big data stream SAVI. Section 3 evaluates the proposed approach TAAM by conducting various simulations and experiments. Section 4 describes the related work, Section 5 discusses limitations and future work, and Section 6 concludes the paper.

2. Adaptive Authenticated Model

The two-phase adaptive authenticated model (TAAM) for big data stream source address validation in SDN-based data center networks includes a data stream collector model, a flow classifier model, and an SPRT-based SAVI model (Figure 1). The sFlow-based data stream collector model can collect real-time network performance information on bandwidth-intensive applications. Depending on the global view of the SDN controller, TAAM can monitor the access/core switches and deploy the conditional entropy-based flow classifier model in the first phase, and then the primary classification of a big data stream is given. In the second phase, we adopt a statistical tool named SPRT to realize a differentiated SAVI model to obtain an extensive analysis of the spoofing possibility of whether a candidate stream requires urgent validation. Table 1 represents the corresponding description of each notation.

2.1. sFlow-Based Data Stream Collector. The data stream collector model (Figure 2) is responsible for big data stream gathering in the access layer. Taking a look at RFC3176[2], sFlow Traffic Collector (sFlow) is a method for monitoring and collecting real-time traffic in a typically switched topology. Sampling, detection, and evaluation were performed by the distributed sFlow agents deployed in switches or routers for getting the sample statistical data in the access layer. We then get continuous SDN-based data center network-wide big data stream information from the collector. The data stream collector model combines an already proposed native OF approach to gather the statistical data stream information in the SDN-enabled switches. Initially, the OpenFlow-based controller encapsulates the OFPT Message in Table 2 into the Packet-in packet (Controller Communication Message) periodically. The periodical operations FEATURES REQUEST are lightweight and easily integrated with existing data stream collection architecture for the separation of the statistical data stream information gathering process from the controller by utilizing the sFlow agents [13]. In detail, TAAM leveraged the packet sampling

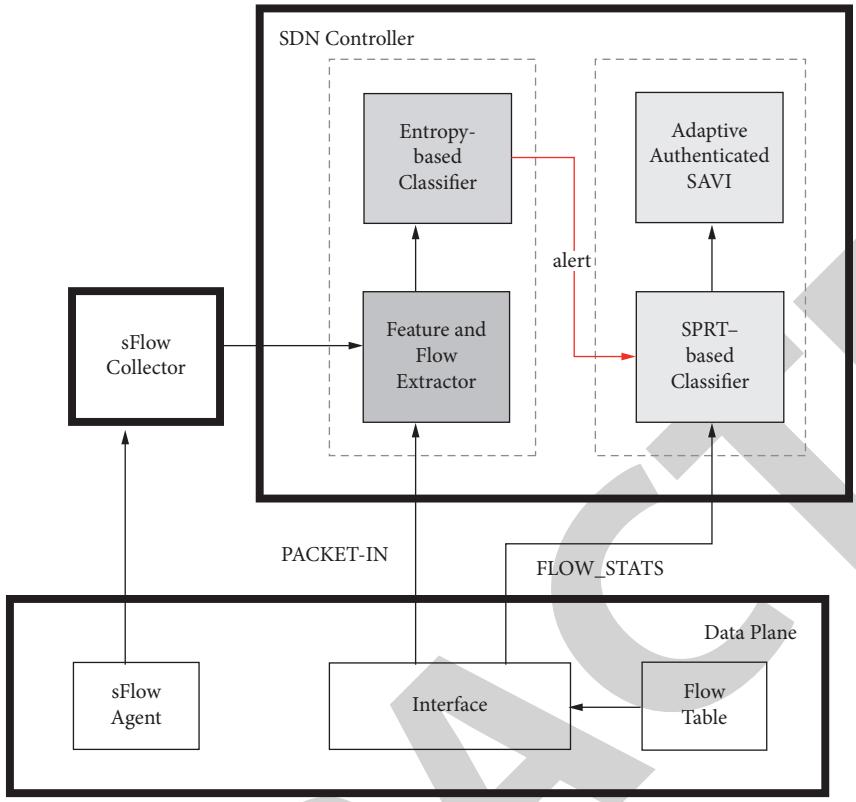

Figure 1: TAAM model design.

Table 1: Notations and descriptions.

Notations	Descriptions
$f_1, f_2, \ldots, f_n$	Different flows events, where n is the total number of occurrences in each predefined window
$d_1, d_2, \ldots, d_n$	Duration time of each flow entry
$f_{i,1}, f_{i,2}, \ldots, f_{i,n}$	Measurements of flow events
$T_1, T_2, \ldots, T_n$	Samples of time intervals in the different flows
E_{flow}	Threshold to classify a candidate flow
H_i	Conditional entropy of match fields
$e_{i,n}$	Conditional entropy of flow at the two consecutive time units of an interval
H_C	Hypothesis that the measurements correspond to all normal/anomaly flow event
H_{A_i}	Anomaly candidate
H_{N_i}	Normal candidate
$l_{A_i,N_i,n}$	SPRT-based dynamic validation at $f_{i,n}$
α	Balance parameters of the false positive error rate
β	Balance parameters of the false negative error rate
λ_0, λ_1	The probability distribution parameters for the flow event

capability of sFlow agent, which decouples entirely the statistical data stream information gathering process from the forwarding logic accompanied by the FEATURES REPLY message (Figure 3). Consequently, the sFlow-based agent is mainly responsible for acquiring the necessary SDN-based data center network information.

Additionally, the sFlow-based data stream collector samples big data stream packets. According to the characteristics of data collection, multithreaded gathering method was used to calibrate the model. The data are generated by the Open vSwitch[3] itself by setting the corresponding parameters obtained from PORT STATUS message on the sFlow agent. Then, the controller sends the view of the current global network topology to the collector. After getting the big data stream information and issuing response actions to related access layer switches, the formatted packet results could be sent to the following models on the controller by UDP protocol. As the sFlow data stream collector receives big data stream packet samples, it updates the corresponding statistical counters inside the sFlow agent. For controller CPU utilization concerns, there is no need to constantly obtain detailed data stream information for each access layer switch of consecutive sampling time windows. Such an approach can reduce the data collection complexity

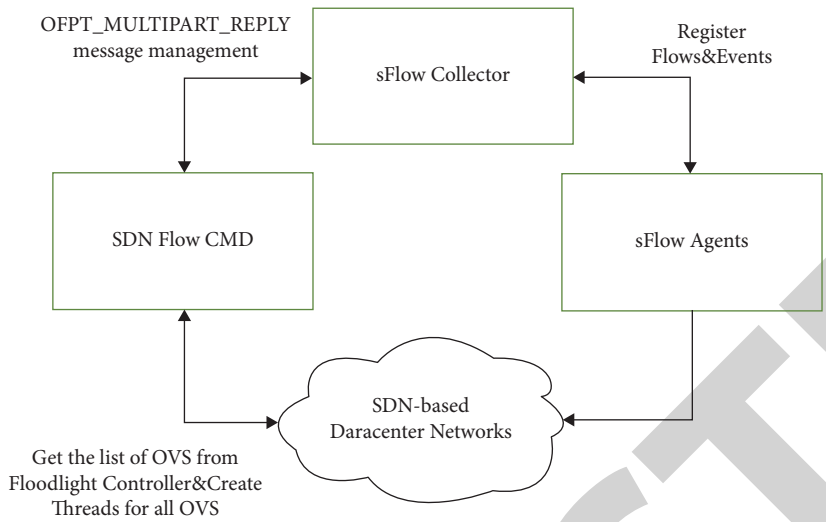

Figure 2: sFlow-based data stream collector model.

Table 2: Typical OFPT messages between the controller and switch.

OFPT message	Communication type	Message description
Hello	Controller to switch	The SDN controller sends its information as identity number to the access layer switch following the corresponding TCP handshake.
Hello	Switch to controller	Supported switch information is replied.
Features request	Controller to switch	The SDN controller requires for validating, which ports on the access layer switch are trusted and available.
Set config	Controller to switch	The SDN controller requires the access layer switch to send flow expired timestamp.
Features reply	Switch to controller	List of flow table match fields (ports, port speeds, supported tables, and actions) is replied by the access layer switches.
Port status	Switch to controller	Enables the access layer switch to inform that SDN controller of flow changes to port speeds or connectivity.

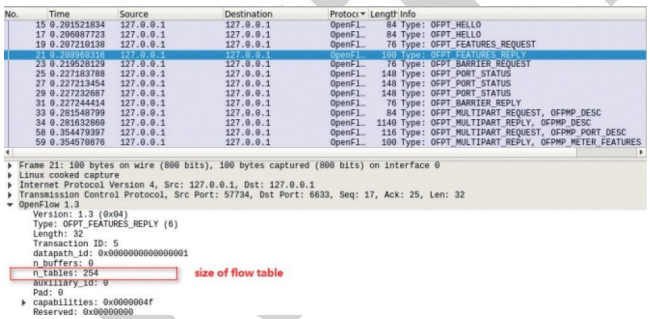

Figure 3: Example of Features reply message.

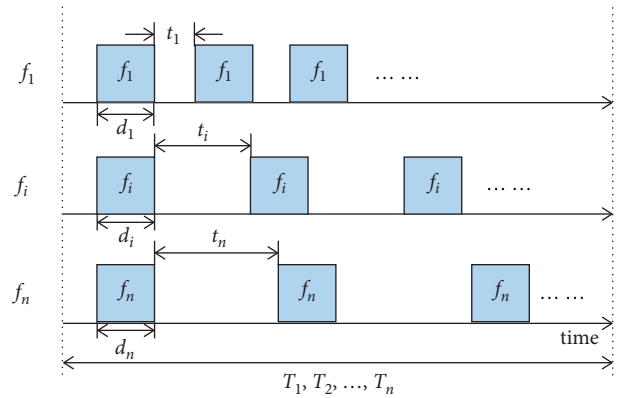

Figure 4: Characteristic parameters of sample flows.

and redundant process of the flow collection algorithm in a certain sampling time window. Meanwhile, it requires lower CPU resources especially when the traffic is growing rapidly. Consequently, the flow features from the sFlow agent are periodically collected and extracted in a big data stream.

2.2. Entropy-Based Flow Classifier. Flows accompanied by big data streams are shown to exhibit variability in terms of duration and interarrival time. As Figure 4 depicts, given that $T_1, T_2, \ldots, T_n$ are the samples of time intervals in the different flows events $f_1, f_2, \ldots, f_n$, and n is the number of samples in the predefined period M, then $d_1, d_2, \ldots, d_n$ are the duration times of each flow entry.

Describing normal and anomaly behaviors is one of the difficulties that an anomaly detection system faces. Shannon's entropy could measure the information content associated with a random variable. Commonly, the higher entropy means the random variable in a certain system with bigger randomness. The data stream feature distribution will

float vastly when the elephant traffic or the spoofing attack happens. According to studies, the launch of a spoofing big data stream attack is usually accompanied by access layer switch traffic exploding [14]. Therefore, it is theoretically possible for preliminaries to classify the flows and corresponding binding relationships in the flow table with suspected spoofed source addresses. Particularly, the threshold value depends on the traffic situation and the distance from the anomalous host to its nearest SDN checkpoint, given that $f_1, f_2, \ldots, f_n$ are the collected flow events. The entropy of a discrete random variable $E(f)$ in the model could be defined as the following equation:

$$E(f) = \sum_{i=1}^{n} -P(f)_i \log_2 P(f)_i, \tag{1}$$

$$H(X \mid Y) = -\sum_{x \in f_{i,X}, y \in f_{i,Y}} p(x \mid y) \log_2 (p(x \mid y)) \tag{2}$$

Conditional entropy is typically defined as the Shannon entropy of conditional circumstances [15]. Four kinds of conditional entropy are defined. Accordingly, four kinds of entropy-based methods for the flow classification in the rough set data analysis are proposed, which is the conditional entropy (equation (2)of match fields, namely, source IP address (sip), destination IP address (dip), destination port (dport), and packet size (psize)) during a time window. Generally, such entropy will change significantly once anomalous conditions happen. As Figures 5 and 6 depict, long-lived flows created by bandwidth-intensive applications incur a large amount of flow rules. Since the concentration of particular addresses under the big data streams, a continuous significance increase in the normalized conditional entropy values (2a) and (2b) could be observed (Figure 5). Consequently, the entropy value changes can significantly reduce the flow table match rate towards the packets arriving in OVS switches. Additionally, the corresponding data counters in the flow tables are also a high probability of unpaired addresses [16]. Only a few packet-in messages generate in the SDN-based network, and there is a sudden decrease in normalized conditional entropy values (2c) and (2d):

$$H'(\text{psize} \mid \text{dport}) = \frac{H(\text{psize} \mid \text{dport})}{\log_2 n}, \tag{2a}$$

$$H'(\text{psize} \mid \text{dip}) = \frac{H(\text{psize} \mid \text{dip})}{\log_2 n}, \tag{2b}$$

$$H'(\text{sip} \mid \text{dip}) = \frac{H(\text{sip} \mid \text{dip})}{\log_2 n} \tag{2c}$$

$$H'(\text{dport} \mid \text{dip}) = \frac{H(\text{dport} \mid \text{dip})}{\log_2 n}, \tag{2d}$$

$$E_{\text{flow}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (H_i - \overline{H})^2}, \tag{3}$$

$$f_{i,n} = \begin{cases} 1, & \text{if } |e_{i,n}| \le E_{\text{flow}}, \\ 0, & \text{if } |e_{i,n}| > E_{\text{flow}}. \end{cases} \tag{4}$$

The real-time conditional entropy changes could be a noteworthy provident to suspect a big data stream is an anomaly and whether a host or port needs to be validated repeatedly [17]. The suspected flow and corresponding binding relationship could be classified by monitoring the conditional entropy value variation persistently. In terms of model implementation, the cache component will record the value and flow events with a timestamp and their timeout. We adopt a successive entropy test component that uses the information stored in the cache to compare the conditional entropy value with a baseline model depicted in equation (3), which is a determined confidence interval of corresponding standard deviation. Let $e_{i,n}$denote the conditional entropy of flow f_n at the two consecutive time units of an interval to be calculated, and let E_{flow} denote the threshold to classify a big data stream. Considering that the collected $f_{i,1}, f_{i,2}, \ldots, f_{i,n}$ within a given window is independent, those candidate flows could be classified as equation (4).

2.3. SPRT-Based Source Address Validation. To realize the source address validation in RFC7513[4], the original SAVI designs a binding-validation model and determines the validation rules in the control plane. Current binding-validation mechanisms in D-SAVI [12] are enforced by flow rules in the access layer switches as in Figure 7, which bind source addresses to switch ports in the <Address, Switch, Port > format. Since the validation rules coexist with the OpenFlow forwarding rules generated by those switches, the latest SDN-based SAVI prototype could use "multiple tables" feature in OpenFlow v1.3 to perform policy-based communications. Therefore, we designed an SPRT-based differentiated SAVI model to verify the suspected spoofing big data streams, which combined the requirements of SAVI efficiency optimization and flow table quantity reduction on the original SAVI basis.

Any flows in normal big data stream originated from the legacy switches will be revalidated to verify their binding relationship state, except that the corresponding flow rules explicitly exist in it [18]. On the implementation side, we adopt a statistical tool named SPRT. The latest SPRT mechanism samples the candidate flows in big data streams and calculates the corresponding likelihood ratio. When confident, it terminates with a spoofed/normal decision. Different from current machine learning approaches that need urgently the feature selection, SPRT-based method is lighter, has fewer features, and is simpler to scale. Particularly, SPRT-based algorithm is easy to be implemented, because it does not depend on a predefined correlation knowledge base or a forehand training of correlation model.

We divide the probability of false positive and false negative into two types as equations (5) and (6). Naturally, the anomaly candidates are mixed with the normal flows. Let H_{N_i} denote the candidate that is observed as a normal candidate, and let H_{A_i} denote the compromised binding relationships with anomaly flows. Let H_C, for

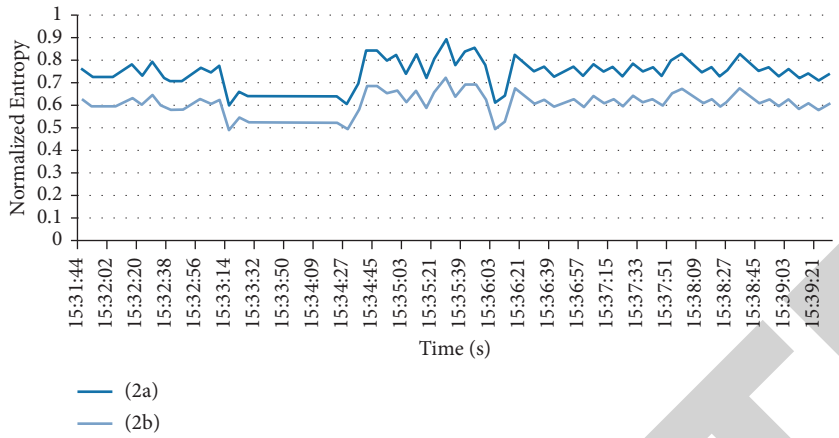

Figure 5: Variation of normalized entropy values (2a) and (2b) under spoofing attack and big data stream.

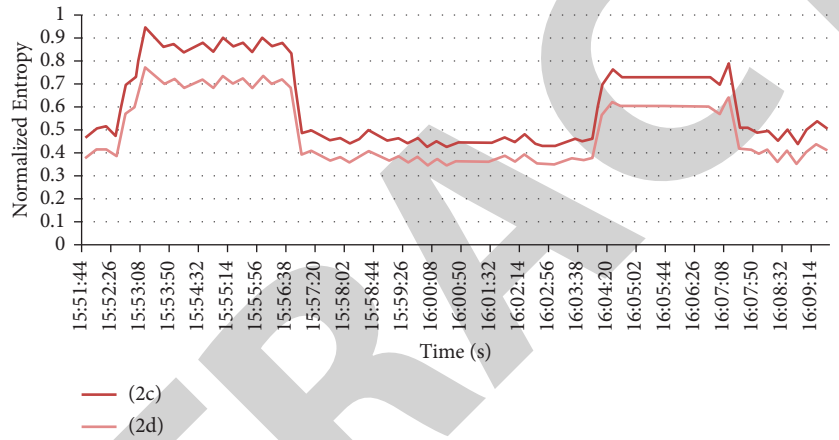

Figure 6: Variation of normalized entropy values (2c) and (2d) under spoofing attack and big data stream.

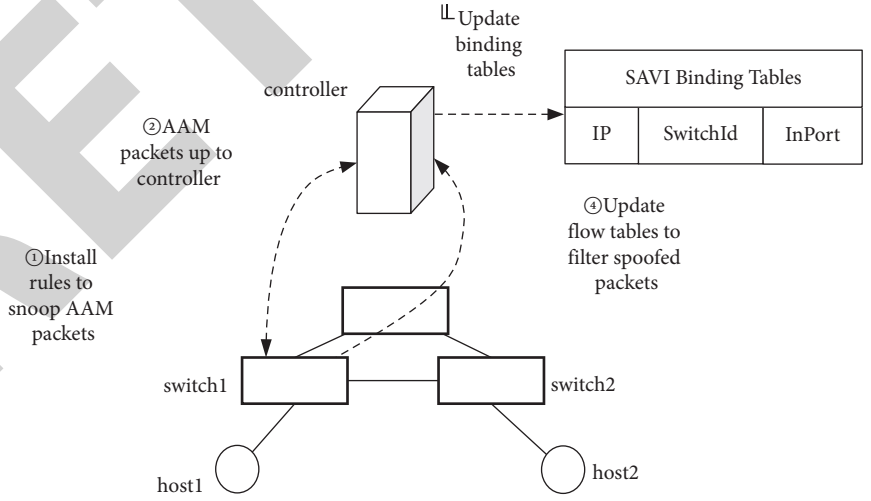

Figure 7: Initial design and binding relationship of SDN-based SAVI.

$C \in \{A_i + N_i, N_i\}$, denote the hypothesis that the measurements correspond to all normal flow and anomaly flow event. False positive means the acceptance of H_C when H_C is true, while the false negative means the acceptance of H_C when H_C is true. To avoid these two errors, α and β are defined as the balance parameters of the false positive and false negative.

Comparing λ_0 with λ_1, the λ_1 is naturally bigger, because a compromised binding relationship is more likely to be injected into the anomaly flows. The approximate maximum

log-likelihood function $L(f_{i,1}, f_{i,2}, \ldots, f_{i,n} \mid H_C)$ could represent the probability ratio of all normal flow and anomaly flow events tested for candidate i. Therefore, we consider $l_{A_i,N_i,n}$ as the following SPRT based on dynamic validation at $f_{i,n}$ as equation (7). Specially, the SPRT-based detection model can be considered as a one-dimensional random walk [19]. By utilizing the entropy-based flow classifiers, we gathered the classification of normal flows and anomaly flows. Therefore, assume that each flow event $f_{i,x}$ is independent and identically distributed, and when a normal flow $|e_{i,n}| \le E_{\text{flow}}$ is tested, we walk upward one step. Otherwise, we walk downward one step. According to equations (5)–(7), we can get equation (8). The intervals of the packet entering the switches will be fed to this module at a periodic time window subsequently. From the above equations, we can conclude that four parameters $\alpha, \beta, \lambda_0, \lambda_1$ are required by the SPRT-based detection method. Among them, α and β limit the false positive error rate and false negative error rate and give two boundaries (A_i and N_i) if the model is considered as a one-dimensional random walk. λ_0, λ_1 are the probability distribution parameters for the flow event $f_1, f_2, \ldots, f_n$ and affect the number of observations required for the dynamic validation to reach a decision.

$$L\left(f_{i,n} = 1 \mid H_{N_i}\right) = 1 - L\left(f_{i,n} = 0 \mid H_{N_i}\right) = \lambda_0, \tag{5}$$

$$\begin{aligned} L\left(f_{i,n} = 1 \mid H_{A_i+N_i}\right) &= 1 - L\left(f_{i,n} = 0 \mid H_{A_i+N_i}\right) \\ &= \lambda_1, \end{aligned} \tag{6}$$

$$l_{A_i,N_i,n} = \ln \frac{L\left(f_{i,1}, f_{i,2}, \ldots, f_{i,n} \mid H_{A_i+N_i}\right)}{L\left(f_{i,1}, f_{i,2}, \ldots, f_{i,n} \mid H_{N_i}\right)}, \tag{7}$$

$$\begin{aligned} l_{i,n} &= \begin{cases} l_{i,n-1} + \dfrac{L\left(f_{i,n} = 1 \mid H_{A_i+N_i}\right)}{L\left(f_{i,n} = 1 \mid H_{A_i+N_i}\right)}, & \text{if } \left|e_{i,n}\right| \le E_{\text{flow}} \\ l_{i,n-1} + \dfrac{L\left(f_{i,n} = 0 \mid H_{A_i+N_i}\right)}{L\left(f_{i,n} = 0 \mid H_{A_i+N_i}\right)}, & \text{if } \left|e_{i,n}\right| > E_{\text{flow}} \end{cases} \\ &= \begin{cases} l_{i,n-1} + \ln \dfrac{\lambda_1}{\lambda_0}, & \text{if } \left|e_{i,n}\right| \le E_{\text{flow}}, \\ l_{i,n-1} + \ln \dfrac{1-\lambda_1}{1-\lambda_0}, & \text{if } \left|e_{i,n}\right| > E_{\text{flow}}. \end{cases} \end{aligned} \tag{8}$$

Above all, the TAAM model averages over all corresponding past estimates to calculate $l_{A_i,N_i,n}$ if the one-dimensional random walk terminates at the i-th global data sample. Specially, when multiple samples in different flow tables are being considered, the calculations will be rather complex. We can utilize all the terminated data samples in the next symbol detection to calculate its likelihood ratio more accurately. The $l_{A_i,N_i,n}$ can be used to classify the candidates with parameters α and β, respectively, as follows:

(1) If $l_{A_i,N_i,n} < \beta/1 - \alpha$, then declare normal big data stream and reduce the D-SAVI intensity, namely, H_{N_i}.

(2) Else if $l_{A_i,N_i,n} > 1 - \beta/\alpha$, then declare that an anomaly big data stream is tested and maintain the D-SAVI intensity, that is, $H_{A_i+N_i}$.

(3) Otherwise, declare that TAAM is not sufficient to make a classification and continue collecting additional statistical access layer switch data.

3. Experiments and Analysis

TAAM is a controller-based model for big data stream SAVI management, which provided an architectural design of a security mechanism. In this section, we implemented a simulated SDN-based data center network to prove TAAM's feasibility and effectiveness. Then, we compare our proposed TAAM with the latest D-SAVI model [12] in terms of performance optimization tests and security ensuring tests.

3.1. Implementation of Experiments. The simulated experimental platform consists of four servers: two servers serve as the host node, and the other servers serve as the simulated edge switches. In the simulation, we use Open vSwitch and Floodlight[5] as the core switches in data center networks and SDN controller, respectively. According to the network resource requirements, simulations are carried out on three typical data center network topologies (Table 3): Abilene (Figure 8), GEANT (Figure 9), and Fat-Tree (Figure 10). The background flow datasets are provided by TOTEM project[6], Uhlig [20], and Fat-Tree [21] on three network topologies correspondingly (Table 3). Mininet 2.3.0[7] is applied for the topology and links simulation for SDN-based data center networks, which supports OpenFlow v1.3 standard. Internet Traffic and Workloads Generator [22] are used to generate the legitimate big data stream. We created a large number of bots by *Python* and Scapy3 to launch the attack, which was carried out in certain parameters as Table 4. We then calculate the number of packets and flows in normal and anomaly traffic on this victim host. As Table 4 depicts, the spoofing attack intensities are divided into three levels. Consequently, the differentiated attack levels of this experimental system can offer scientific quantitative analysis and appraisal to source address validation model.

3.2. Parameter Tests. Table 5 provides the detection rate of the two-phase algorithm applied in TAAM. Initially, the benchmark of the two-phase algorithm applied in TAAM was calculated. To evaluate the performance of SPRT-based source address validation algorithm, we compared it with six other classic machine learning algorithms including XGBoost in D-SAVI [23]. The choice of entropy-based and SPRT-based algorithms has its advantage in model training time while retaining most of the performance benefits including precision and recall. Particularly, SPRT-based big data stream classification algorithm to realize differentiated verifications, with no additional machine learning model training time, does not take up real-time controller memory space.

Aiming at choosing the upper and lower thresholds, respectively, as shown in Figure 11, we further tested the

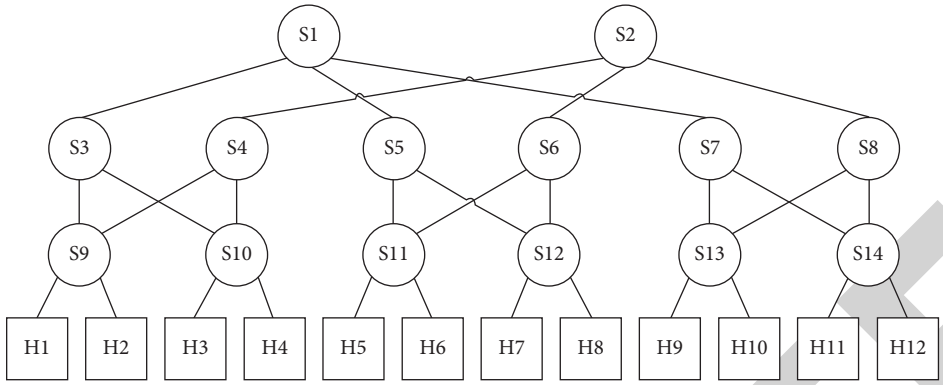

Figure 8: Abilene SDN-based data center network topology.

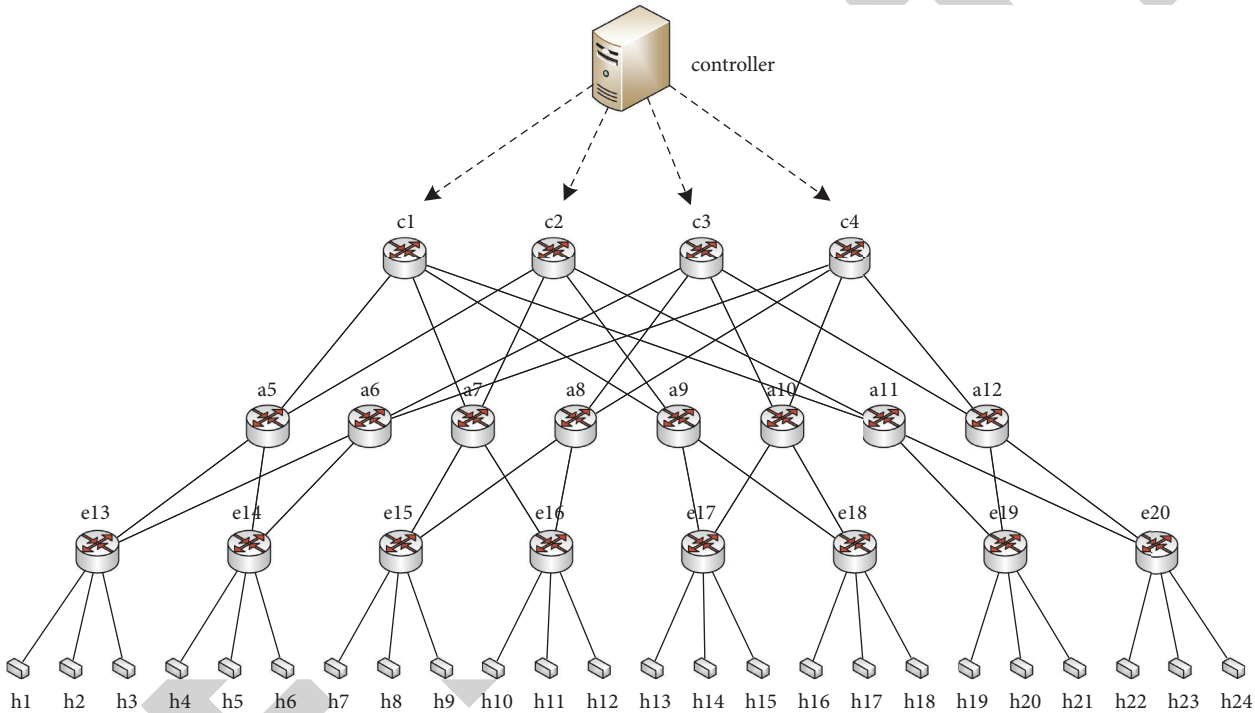

Figure 9: GEANT SDN-based data center network topology.

influence of the threshold values towards the number of successive tests in the aforementioned SPRT-based differentiated validation model (Section 3.3). Basically, the greater the difference between λ_0 and λ_1, the smaller the successive number of tests required for our method to reach detection. Furthermore, the detailed value of λ_0 and λ_1 in our evaluation could be also shown in Figure 12, and it can be concluded that our method can detect a compromised candidate after 6 to 8 successive tests. There is a similar trend for the average number of required validations. However, such a test mechanism is not exactly implementable since the upper and lower threshold values now depend on the random parameters in different data center network topologies.

We then summarize the normal candidates' proportion in all the binding relationships for training data. Under background big data stream traffic in Figure 13 for three different SDN-based data center topologies, and α and β are naturally small values limiting the false positive rate and false negative rate, which are usually between 0.01 and 0.05. Since the process of OVS flow table update operation is hard and very slow when the flow table load is too large, α and β are defined as the balance parameters of the false positive and false negative to avoid these two errors.

Here, we set $\alpha = 0.01$ and $\beta = 0.02$ in the evaluation and estimate λ_0 and λ_1. Afterwards, we calculated the average abnormal candidates' proportion (Figure 13) to estimate λ_0 and λ_1. As Table 6 depicts, different network environments make different λ_0 and λ_1 values. In Section 3.4, we have mentioned that the likelihood function $L(f_{i,1}, f_{i,2}, \ldots, f_{i,n} \mid H_C)$ could represent the probability ratio of all normal flow and anomaly flow events tested for candidate i. The results show that, in the different topology of Abilene, GEANT, and Fat-Tree, λ_1 is indeed bigger, because an anomaly candidate under big data stream is more likely to be injected into the anomaly flows. Another

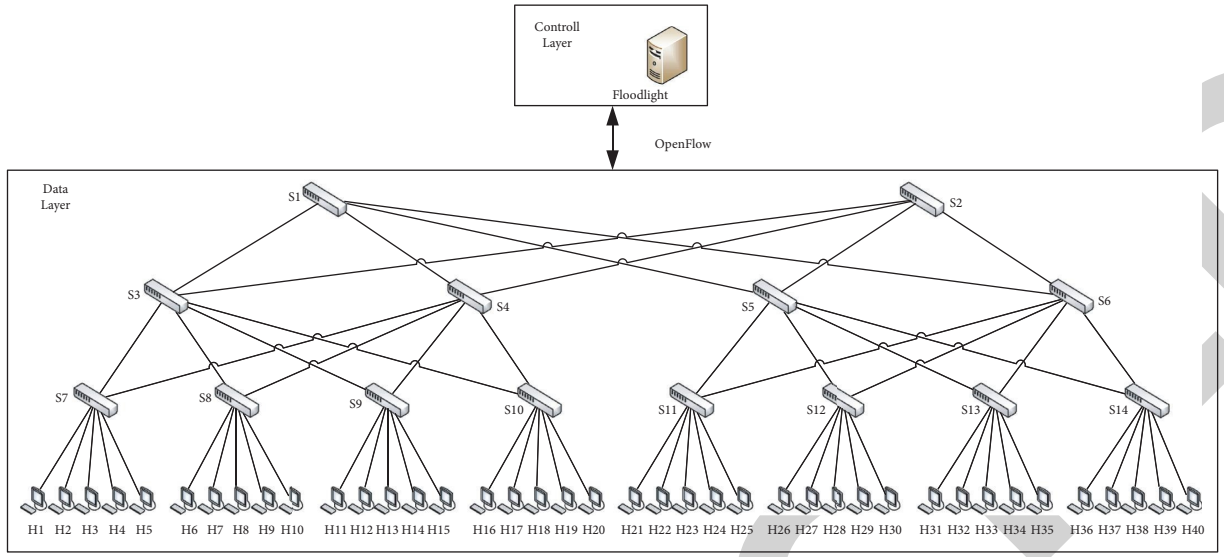

Figure 10: Fat-Tree SDN-based data center network topology.

Table 3: Scale of three simulated SDN-based data center networks.

Topology ID	Topology name	Basic configuration in experimental topology		
		Details	Nodes	Links
1	Abilene (Figure 8)	Attackers Nodes:4 server Nodes:8	12	30
2	GEANT (Figure 9)	Attackers Nodes:8 server Nodes:16	24	56
3	Fat-tree (Figure 10)	Attackers Nodes:10 server Nodes:30	40	64

Table 4: Descriptions and benchmarks of the big data stream attacks

Attack nodes	Normal packets (packet/s)	Spoofing packets (packet/s)	Attack descriptions	Type	Spoofing packets		Spoofing flows	
					Normal	Anomaly	Normal	Anomaly
$H1$	5×10^4	1×10^5	**79.3%SYN flood, 15.9%UDP flood, 4.2%TCP flood, 0.6%ICMP flood**	(A)	10561183	1584185	794403	185826
$H2$	5×10^4	2×10^5						
$H3$	5×10^5	3×10^5		(B)	5871867	5861451	591788	212512
$H4$	5×10^5	4×10^5						
$H9$	2×10^6	5×10^5						
$H10$	2×10^6	6×10^5		(C)	2598123	12990259	316125	386521
$H12$	5×10^6	8×10^5						

Table 5: Metrics of performance for different algorithms.

Algorithm	Precision	Recall	Training time (sec)
TAAM	**0.9526**	**0.9812**	**0**
XGBoost	0.9751	0.9875	14.723
DT	0.9604	0.9733	28.452
SVM	0.7847	0.9901	67.417
KNN	0.9574	0.9518	93.792
RF	0.9316	0.6126	4.875
NB	0.7966	0.9492	3.538

important observation is that the difference between λ_0 and λ_1 is bigger, accompanied by the growth of the topology scale.

3.3. Performance and Overhead Tests. Regarding performance and overhead tests, we provide results under different network types and topologies. A recent set of comparisons in Table 7 imply that the D-SAVI has the edge on performance for now, but that our proposed TAAM promises better performance across the board due to the reduction in redundant processes of source address validation (26.2%, 36.4%, and 42.9%). Furthermore, TAAM guards against the spoofing attack with the integrated use of entropy-based flow classifier and SPRT-based dynamic validation to evaluate real-time big data stream conditions of the SDN-based data

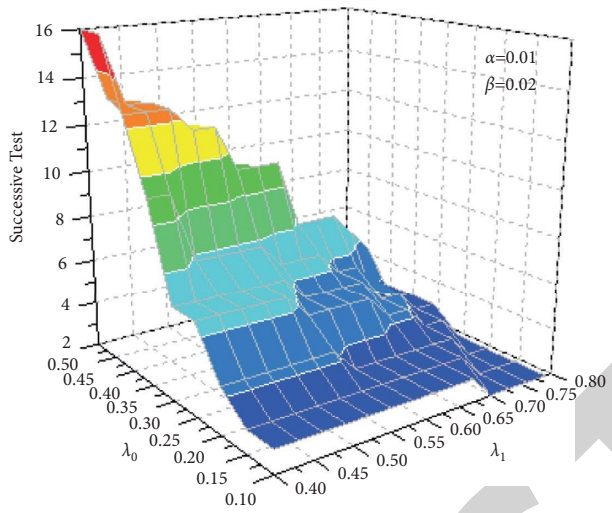

Figure 11: Minimum number of successive SPRT tests for detecting an anomalous candidate.

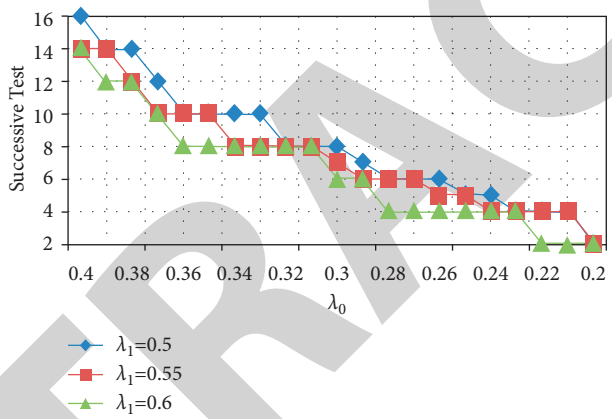

Figure 12: Detailed numbers of successive SPRT tests.

center network. These results also imply that the TAAM model is more effective than other existing methods to reduce average packet forwarding delay with limited anomalies. Compared with the original OpenFlow, the experimental results demonstrate the effectiveness of SAVI deployment in TAAM and show an obvious optimization in high bandwidth and large traffic environments.

Consequently, our proposed TAAM model ensures more significant average packet forwarding delay optimization as the scale of the simulated data center network gradually expanded. Considering that the flow table update rate is effectively reduced Table 7, it also has the advantages of stability and reliability. TAAM is superficial in the aspect of average packet forwarding delay, because the differentiated SPRT-based source address validation can resist big data stream and simplify the executing process of binding relationship verification to reduce the traffic cost. However, the apparent advantage may narrow because the random polling module starts its security guarantee measures under attack Type C (80% Spoofing Packets). Furthermore, the existence of the SPRT-based dynamic validation model is the main factor that affects the SAVI system performance. For different topology sizes, the average packet forwarding delay for TAAM performs better than the other techniques. The reason is that a differentiated validation model is a technique that optimizes the original polling mechanism to reduce latency and the candidate validation frequency.

Additionally, by measuring the controller's CPU utilization, the TAAM model overhead could be evaluated (Figure 14). A serial CPU utilization rate effect was found when the entropy-based methods were used. Corresponding results show that the average controller CPU utilization is 8% without TAAM and 10.7% with TAAM deployment, which indicates that TAAM incurs little overhead under normal states. Consequently, TAAM ensures reliable identification of incoming packets substantially reducing the risk of accidentally filters on a normal host. The average controller's CPU utilization value proves that our proposed model, which can assure the safety and highly effective SDN-based data center network management, is safe, secure, and effective.

4. Related Works

A reliable source address validation is a significant prerequisite for big data stream communication and

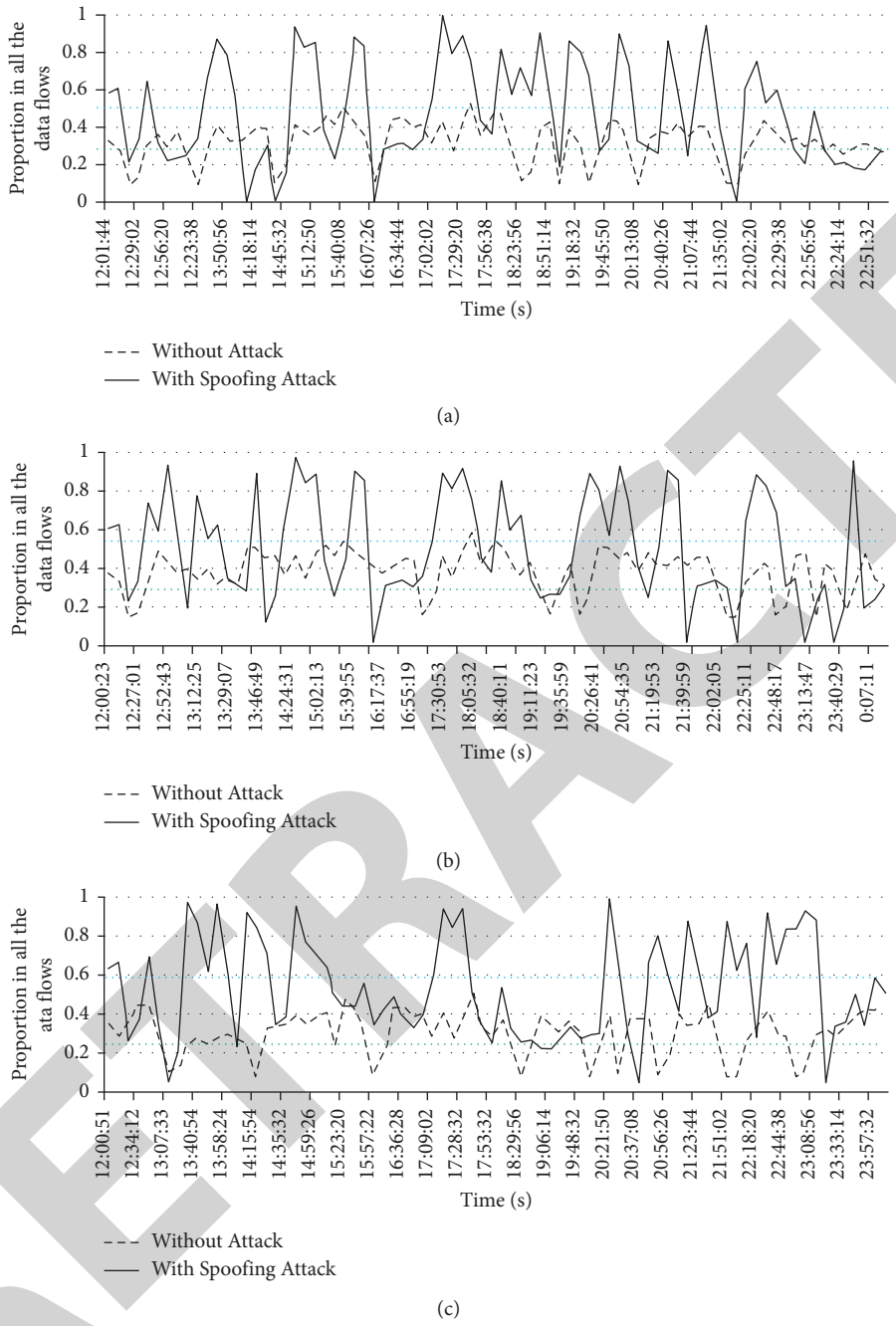

Figure 13: Proportion of the anomaly candidates. (a) Alilene. (b) GEANT. (c) Fat-Tree.

Table 6: Estimation value of λ.

Topology	λ_0	λ_1
Abilene	0.29	0.51
GEANT	0.27	0.56
Fat-Tree	0.24	0.59

authentication in an SDN-based data center network. Aiming at the spoofing source address attack problem, the working group of IETF is first to standardize the prototype system of source address validation implementation (SAVI). SAVI sets up a Layer-2 switch and filter spoofing packets by establishing a binding-validation model for data transmission and communication [24]. Such a binding-validation mechanism increases the filtering granularity in SAVI. Meanwhile, SDN simplifies the application management by utilizing various policy-based controls over SDN-enabled

Table 7: Performance tests of the authenticated model under big data stream.

Type		Average packet forwarding delay (ms)			Optimization percentage	
		OPENFLOW	D-SAVI	TAAM	Over OPENFLOW (%)	Over D-SAVI (%)
A	1	1898 ± 83	2510 ± 142	1852 ± 69	2.4	**26.2**
	2	2172 ± 74	3525 ± 196	2241 ± 107	3.2	**36.4**
	3	3553 ± 112	5378 ± 287	3072 ± 167	13.5	**42.9**
B	1	2184 ± 213	2461 ± 275	2078 ± 142	4.9	15.6
	2	3188 ± 173	3272 ± 372	2685 ± 192	15.7	17.9
	3	4884 ± 247	5060 ± 249	3907 ± 274	20.1	22.7
C	1	4632 ± 275	3917 ± 468	3725 ± 240	**19.5**	4.9
	2	6230 ± 309	5213 ± 338	4867 ± 239	**21.8**	6.6
	3	9512 ± 321	7932 ± 292	6961 ± 261	**26.8**	12.2

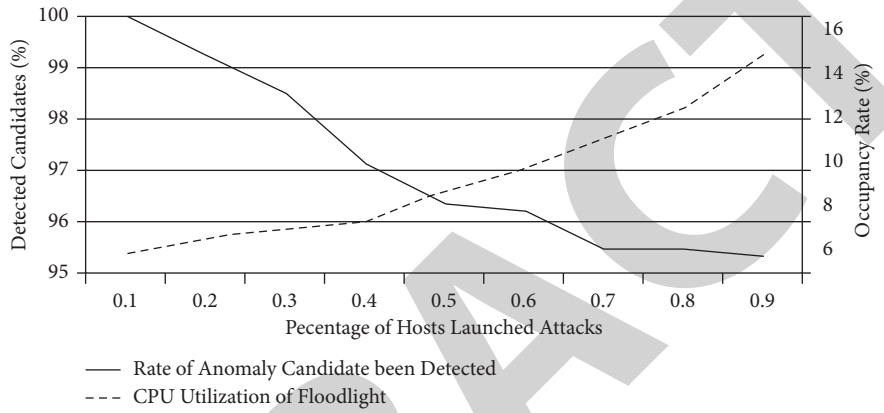

Figure 14: Model performance of TAAM under different spoofing attack percentage.

switches. The scalability of authenticated model implementation is single hosts rather than IP prefixes in the source addresses, which is much more accurate than conventional methods.

SAVI Filtering IP spoofing traffic with agility (VASE) and Virtual Source Address Validation Edge (VAVE) are both implementations under SAVI [25]. The purpose of the VASE and VAVE is to protect users in the communication systems from being spoofed by attackers within the same network domain. VASE and VAVE establish an SAVI protection zone comprising all of the communication devices including the Layer-3 OpenFlow switches and L2 SAVI switches. However, few of these papers discuss how to implement SAVI technology based on the SDN-based data center networks. On the other hand, based on the convenience provided by the SDN northbound programmable interface, BGP-based Anti-Spoofing Extension (BASE) [6] is an anti-spoofing protocol based on SDN specifications, and the implementation will be the existing authentication technology for hybrid SDN-based networks. Source Address Validation in Software Defined Networks (SDN-SAVI) [26] was the preliminary design to enable IPv6-based SAVI functionalities and implementation under SDN deployment. Benefited from the global view of SDN controller, SDN-SAVI deploys authentication technology through IPv6-based flow tables installed in the access layer switch without implementing redundant settings. SDN-SAVI is excellent for externalizing configuration settings that may need to be changed by the network manager.

To ensure the source address authenticity and security in the Content Delivery Networks (CDN), the authors in [27] proposed a mechanism CDNi that can detect spoofed source addresses and therefore create a robust defense against spoofing attack created by spoofed IP. However, one obvious shortcoming is that the current CDNi is unable to prevent users in the corresponding communication systems from forging the source addresses in the same network domain. Source Address Validation for SDN Hybrid networks (SAVSH) [28] can locate spoofed nodes for the OpenFlow switch replacement and deploy filtering flow rules and authentication code onto them with a desirable fine-grained administrative security level. SAVSH only takes a few SDN authentication devices as possible but trades desirable extensibility of the authentication tools and of the development workbench.

SDN-based Integrated IP Source Address Validation Architecture (ISAVA) computes the [29] forwarding path in a local scope of the network for the problem of source address anonymity protection. To ensure that the source address generated after the execution of communication systems can uniquely identify the current big data stream, ISAVA proposes an authenticated data structure with privacy-preserving based on a longer verification path. Such paths enable a series of changes of the source address in packets of a big data stream, which makes it is suitable for SDN-based big data stream scenarios. SDN-Ti [30] is proposed for tracing back and identifying spoofing attackers in SDN-based data center networks. Switches applying SDN-Ti

could extend the functionality of the SDN switch and controller, and it is also used in router of access networks. SDN-Ti is intelligent at recognizing the spoofing devices and quick at snooping address configuration packets. However, there still exist some drawbacks in a big data stream scenario.

5. Limitations and Future Works

To the best of our knowledge, this study is the first work to discuss how to validate source addresses using a differentiated SAVI mechanism under big data streams. However, the TAAM has more or fewer limitations when used in an SDN-based data center network. In this section, we discuss some of the limitations.

Initially, to defend against source address spoofing attack in big data streams, the SPRT-based source address validation model must last for a time window and test different approaches to maintain the security level. However, setting up a conditional time window in the access switches means that every new flow in the big data stream has to be classified repeatedly, which could put extra pressure on the adaptive authenticated model. Furthermore, the topology of the SDN-based data center can affect the flow classifier. For instance, an SDN controller in the data center network may be directly attacked by several sophisticated spoofing attackers coordinately, making the entropy-based classifier less effective. Alternatively, the adaptive authenticated model can make much greater use of the SDN controller's global view to calculate an entropy-based threshold of the limitation.

While the present results of model overhead and performance tests have verified the efficiency of the adaptive authenticated model, some flaws underlaid in the adaptive authenticated model construction could not meet specific traffic requirements completely. As a next step, we will attempt to implement the analytical model to a real data center network as a more practical model. Besides, it is significant to find adaptive methods to differentiate the spoofing packets in the big data stream from legitimate ones in real-time before they enter further into the control layer, therefore solving the IP source address validation problem in SDN-based data center network fundamentally. Our future work will mainly focus on further optimization of anomaly detection techniques and try to deploy it in other actual SDN-based data center networks.

6. Conclusion

This paper presented an adaptive authenticated model for big data stream SAVI in SDN-based data center networks. We propose a two-phase adaptive authenticated model by introducing SAVI-based IP source address verification. Our proposed TAAM model realizes real-time verification of data stream and dynamically reduces the redundant big data stream SAVI process. The model overhead and performance test results demonstrate that the two-phase adaptive authenticated model achieves desirable security, efficiency, and stability.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant no. 2020YFB1805601, National Network Security Key Research and Development Program of China under Grant no. 2017YFB0801703, and the CERNET Innovation Project under Grant no. NGII20170408.

References

[1] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *Computers & Chemical Engineering*, vol. 91, no. 2, pp. 182–194, 2016.

[2] Y. Wang, X. Wang, H. Li, Y. Dong, Q. Liu, and X. Shi, "A multi-service differentiation traffic management strategy in SDN cloud data center," *Computer Networks*, vol. 171, Article ID 107143, 2020.

[3] W. Wang, M. Dong, K. Ota, J. Wu, J. Li, and G. Li, "CDLB: a cross-domain load balancing mechanism for software defined networks in cloud data centre," *International Journal of Computational Science and Engineering*, vol. 18, no. 1, 2019.

[4] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow," *ACM SIGCOMM - Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] M. H. Wang, J. W. Guo, C. L. Lei, and P. W. Chi, "MigrateSDN: efficient approach to integrate OpenFlow networks with STP-enabled networks," *International Journal of Computational Science and Engineering*, vol. 20, no. 4, p. 480, 2019.

[6] J. Kwon, D. Seo, M. Kwon, H. Lee, A. Perrig, and H. Kim, "An incrementally deployable anti-spoofing mechanism for software-defined networks," *Computer Communications*, vol. 64, no. 15, pp. 1–20, 2015.

[7] G. Yao, J. Bi, and A. V. Vasilakos, "Passive IP traceback: disclosing the locations of IP spoofers from path backscatter," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 471–484, 2015.

[8] C. Liu and P. Xiao, "A novel virtual disk bandwidth allocation framework for data-intensive applications in cloud environments," *International Journal of Computational Science and Engineering*, vol. 20, no. 1, p. 21, 2019.

[9] O. Hosam and M. H. Ahmad, "Hybrid design for cloud data security using combination of AES, ECC and LSB steganography," *International Journal of Computational Science and Engineering*, vol. 19, no. 2, p. 153, 2019.

[10] Y. Sun, Q. Liu, X. Chen, and X. Du, "An adaptive authenticated data structure with privacy-preserving for big data stream in cloud," *IEEE Transactions on Information Forensics*, vol. 99, p. 1, 2020.

[11] M. Bagnulo and A. Garcia-Martinez, "SAVI: the IETF standard in address validation," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 66–73, 2013.

[12] Q. Zhou, J. Yu, and D. Li, "A Dynamic and Lightweight Framework to Secure Source Addresses in the SDN-Based Networks," *Computer Networks*, vol. 193, 2021.

[13] M. Cicioglu and A. Çalhan, "HUBsFLOW: a novel interface protocol for SDN-enabled WBANs," *Computer Networks*, vol. 160, pp. 105–117, 2019.

[14] O. Runsewe and N. Samaan, "Cloud resource scaling for time-bounded and unbounded big data streaming applications," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 504–517, 2021.

[15] H. Wang and D. You, "Online streaming feature selection via multi-conditional independence and mutual information entropy†," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 479–487, 2020.

[16] H. Mahmood, D. Mahmood, Q. Shaheen, R. Akhtar, and W. Changda, "An SDN-based DDoS protection system for smart grids," *Security and Communication Networks*, vol. 2021, Article ID 6629098, 19 pages, 2021.

[17] Y. Xu, Y. Yu, H. Hong, and Z. Sun, "DDoS detection using a cloud-edge collaboration method based on entropy-measuring SOM and KD-tree in SDN," *Security and Communication Networks*, vol. 2021, Article ID 5594468, 16 pages, 2021.

[18] Y. Afek, A. Bremler-Barr, and L. Shafir, "Network anti-spoofing with SDN data plane," in *Proceeding of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, USA, May 2017.

[19] D. Ping, X. Du, H. Zhang, and X. Tong, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," in *Proceeding of the IEEE International Conference on Communications*, Kuala Lumpur, Malaysia, May 2016.

[20] A. Liakopoulos, B. Maglaris, C. Bouras, and A. Sevasti, "Providing and verifying advanced IP services in hierarchical DiffServ networks-the case of GEANT," *International Journal of Communication Systems*, vol. 17, no. 4, pp. 321–336, 2004.

[21] W. Li, Y. Xu, K. Li, H. Qi, and X. Zhou, "Leveraging endpoint flexibility when scheduling coflows across geo-distributed datacenters," in *Proceedig of the IEEE International Conference on Computer Communications*, vol. 2018, April 2018.

[22] G. Aceto, A. Botta, W. D. Donato, and A. Pescapè, "D.-I. T. G.: Distributed internet traffic generator," *Praxis der Informationsverarbeitung und Kommunikation*, vol. 36, no. 1, p. 49, 2013.

[23] T. Chen, H. Tong, and M. Benesty, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, August 2016.

[24] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in *Proceeding of the IEEE International Conference on Network Protocols*, Vancouver, Canada, October 2011.

[25] J. Li, J. Bi, and J. Wu, "Towards a cooperative mechanism based distributed source address filtering," in *Proceeding of the International Conference on Computer Communications & Networks*, pp. 1–7, Nassau, Bahamas, August 2013.

[26] B. Liu, J. Bi, and Z. Yu, "Source address validation in software defined networks," in *Proceeding of the Acm Sigcomm Conference*, Florianopolis Brazil, August 2016.

[27] N. I. Mowla, I. Doh, and K. Chae, "An efficient defense mechanism for spoofed IP attack in SDN based CDNi," in *Proceeding of the International Conference on Information Networking*, vol. 2015, January 2015.

[28] G. Chen, G. Hu, Y. Jiang, and C. Zhang, "SAVSH: IP source address validation for SDN hybrid networks," in *Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC)*, Messina, Italy, June 2016.

[29] R. Guerzoni, R. Trivisonno, and D. Soldani, "SDN-based architecture and procedures for 5G networks," in *Proceeding of the International Conference on 5g for Ubiquitous Connectivity*, Akaslompolo, Finland, November 2014.

[30] C. Li, Q. Wu, H. Li, and J. Zhou, "SDN-Ti: a general solution based on SDN to attacker traceback and identification in IPv6 networks," in *Proceeding of the ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019.