Hindawi
Security and Communication Networks
Volume 2023, Article ID 9856153, 1 page
https://doi.org/10.1155/2023/9856153

Retraction

Retracted: DDoS Detection Using a Cloud-Edge Collaboration Method Based on Entropy-Measuring SOM and KD-Tree in SDN

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This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1].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] 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.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5594468, 16 pages
https://doi.org/10.1155/2021/5594468

Research Article

DDoS Detection Using a Cloud-Edge Collaboration Method Based on Entropy-Measuring SOM and KD-Tree in SDN

Yuhua Xu,[1] **Yunfeng Yu**,[2] **Hanshu Hong**,[1] **and Zhixin Sun**[1]

[1]*Engineering Research Center of Post Big Data Technology and Application of Jiangsu Province, Research and Development Center of Post Industry Technology of the State Posts Bureau (Internet of Things Technology), Engineering Research Center of Broadband Wireless Communication Technology of the Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2]*Guoji Beisheng (Nanjing) Technology Development Co., Ltd, Nanjing 210003, China*

Correspondence should be addressed to Zhixin Sun; sunzx@njupt.edu.cn

Received 10 February 2021; Revised 3 March 2021; Accepted 25 March 2021; Published 12 April 2021

Academic Editor: Chi-Hua Chen

Software-defined networking (SDN) emerges as an innovative network paradigm, which separates the control plane from the data plane to improve the network programmability and flexibility. It is widely applied in the Internet of Things (IoT). However, SDN is vulnerable to DDoS attacks, which can cause network disasters. In order to protect SDN security, a DDoS detection method using cloud-edge collaboration based on Entropy-Measuring Self-organizing Maps and KD-tree (EMSOM-KD) is designed for SDN. Entropy measurement is utilized to select the ideal SOM map and classify SOM neurons considering the limitation of dead and suspicious neurons. EMSOM can detect most flows directly and filter out a few doubtable flows. Then these flows are fine-grained, identified by KD-tree. Due to the limited and precious resources of the controller, parameter computation is performed in the cloud. The edge controller implements DDoS detection by EMSOM-KD. The experiments are conducted to evaluate the performance of the proposed method. The results show that EMSOM-KD has better detection accuracy; moreover, it improves the KD-tree detection efficiency.

1. Introduction

Software-defined networking (SDN) separates the control plane from the data plane to achieve programmable, flexible, and reliable network services [1]. In recent researches [2, 3], SDN combined with edge computing is applied in the Internet of Things (IoT) such as smart city and ubiquitous healthcare. Edge controllers of SDN implement logically centralized management of the local data plane and collect the network information from forwarding devices to maintain a global view of the local network [4]. According to the flow table, forwarding devices such as switches forward data packets in the data plane. OpenFlow protocol is widely used between the data plane and the control plane [5].

However, SDN is vulnerable to Distributed Denial of Service (DDoS) attacks due to its centralized control framework. A large number of malicious packets with spoofing addresses being sent to switches can easily lead to buffer saturation and flow table overflow [6] because switches in the data plane have limited resources. What is more, switches are forced to send numerous packet_in messages to the controller for flow requests. This forms packet_in flooding on the controller and causes the controller saturation [7]. Therefore, DDoS attacks can lead to network collapse, and flow detection is essential for SDN network security.

Various algorithms are applied as classifiers for flow identification in SDN. The performance of different algorithms can affect the effectiveness of DDoS defense in SDN. The self-organizing map (SOM) is one of the most effective classifiers [8], and it can efficiently classify SDN flows. SOM maps high-dimensional training data to low-dimensional winning neurons of the neural network and recognizes network flows through winning neurons [9]. However, SOM

neural network which is not set automatically can affect the detection accuracy. There are dead neurons that have never been mapped by training data and suspicious neurons that map similar numbers of normal and abnormal training data. These neurons lower the detection precision of SOM. High-precision flow detection can protect the communication security of SDN. *K*-Nearest Neighbor (KNN) has high detection accuracy of DDoS detection [10]. However, its high time-consuming leads to detection delays and puts tremendous pressure on the controller. Therefore, an efficient and accurate detection method is essential for DDoS defense in SDN. Moreover, parameters calculation of the detection method can increase the centralized controller overhead and compromise the controller performance.

SDN DDoS detection framework can be divided into two main modes. In the first mode, the smart DDOS detection algorithm, such as the deep learning algorithm [11, 12], is deployed in the controller. However, the smart algorithm training process can significantly impact the controller and make the controller be the network bottleneck. In the second mode, the lightweight algorithm, such as the entropy-based algorithm, is used by switches [13] to detect abnormal flows and share the controller workload. But switches have limited computing and storage resources, and additional detection workload of switches may affect network communication. Unlike previous researches, we propose a cloud-edge collaboration DDoS detection method. The cloud server computes the parameters and implements the training process of the improved smart algorithm to reduce the burden on the controller. The controller can detect DDoS attacks efficiently and accurately by the improved algorithm combining Entropy-Measuring SOM and KD-tree. Our contributions are summarized as follows:

(1) A cloud-edge collaboration DDoS detection framework is designed for SDN. It decouples parameter calculation from flow detection. The cloud performs the detecting parameter calculation, and it helps the edge controller focus on traffic detection to reduce the workload.

(2) A detection method based on Entropy-Measuring SOM and KD-tree (EMSOM-KD) is proposed to efficiently and precisely detect network traffic. A scoring scheme is built by the entropy measurement to compute a suitable SOM map, which can classify flows precisely and filter out a small number of suspicious flows. Then, KD-tree is utilized for the identification of suspicious flows.

(3) The experiments are made in detail to verify the proposed method's effectiveness and efficiency.

The remainder of this paper is organized as follows: the related work of DDoS detection for SDN is introduced in Section 2. Section 3 introduces a cloud-edge collaboration framework for DDoS detection in SDN and presents the details of EMSOM-KD detection algorithm. In Section 4, experiments are conducted for the performance evaluation of the proposed method. Section 5 concludes the paper and points out the future work.

2. Related Work

DDOS detection solutions for SDN networks can be mainly divided into statistical solutions, machine learning-based solutions, and artificial neural networks-based solutions.

The statistical detection solutions monitor and count the flow information of SDN and then compares the statistical value with the threshold to determine whether the traffic is an attack. Fouladi [14] and Bawany [15] used filters and set dynamic thresholds to detect instant abnormal changes. Sahoo et al. [16–18] utilized information entropy-based methods to detect DDoS attacks in the control plane. Although the statistics-based scheme is efficient and straightforward, the threshold value setting that requires multiple statistics is difficult.

Machine learning solutions use clustering algorithms, decision tree, SVM, KNN, etc., as classifiers to determine the DDoS attacks. Cui [19] computed the dual address entropy as the main feature and utilized SVM to detect flows. In the research [20], a whale optimization algorithm is proposed for DDoS detection in SDN. Chen [21] modified the decision tree algorithm to detect the SDN network state. Latah [22] compared KNN with other machine learning algorithms and showed that KNN has high detection precision. Tuan [23] and Dong [24] deployed the KNN-based detector in the controller for high-precision anomaly detection. But the traditional KNN needs to calculate the distance between the detection point and each training point and causes significant detection delay. To reduce the calculation time of traditional KNN, *k*-dimensional (KD) tree was established [25] to realize rapid search of the nearest k points while maintaining the accuracy of KNN. However, the detection speed of KD-tree still needs to be further improved to realize efficient DDoS detection in SDN.

Solutions of artificial neural networks (ANN) simulate the human brain structure to abstract knowledge through automatic learning for flow identification [26]. Hannache [27] proposed a Neural Network based Traffic Flow Classifier (TFC-NN) to detect DDoS attacks in the SDN environment. Han [28] combined autoencoder and softmax classifier for DDoS detection. The complex training process of ANN puts computational pressure on the controller. As one of ANN algorithms, SOM trains the neurons to form the SOM map whose different units represent different traffic types [29]. Because of its efficient classification capability, SOM is widely used for DDoS detection in SDN. Trung [30] designed a distributed SOM for flooding attacks. Tran [31] combined SOM with KNN for the improvement of SOM detection accuracy. The topological structure of the SOM map, which is not automatically set, can affect the detection results. Thus it needs to be improved for detection precision.

Based on the comprehensive analysis above, an efficient and accurate DDoS detection method is the key to DDoS defense in SDN. The parameter calculation of the detection algorithm consumes the controller resources and affects its performance. Therefore, we design a cloud-edge collaboration architecture to strip the preprocessing calculation of DDoS detection from the controller and improve the

existing detection method to realize efficient and accurate flow identification.

3. Cloud-Edge Collaboration Detection System Based on EMSOM-KD

The SDN controller communicates directly with switches and obtains a global network topology. Furthermore, it has a centralized network operating system facilitating anomaly detection and mitigation [32]. However, the centralized computation also puts much pressure on the controller. The proposed hierarchical detection architecture separates the detection parameter calculation from the flow detection to reduce the controller burden, as shown in Figure 1. The cloud server calculates the detection parameters and deploys them in the edge controller. The resources of complex parameter calculations in the controller can be freed up. Therefore, the edge controller can focus on lightweight traffic detection.

Each switch stores the flow table of OpenFlow protocol for network flow forwarding. The flow table has a set of flow entries consisting of header fielders, counters, and actions [33]. The edge controller connects with switches to collect flow information, detects flows by the detection method based on EMSOM-KD, and mitigates DDoS attacks by setting actions in the flow table. The proposed detection framework is shown in Figure 2, the preprocess modules are in the cloud server, and detection modules are in the edge controller.

3.1. Preprocess Modules in Cloud Server. Preprocess modules in the cloud server include Database, KD-tree Builder, EMSOM Preprocessor, and EMSOM-KD Parameter Transmitter. They implement computation and transmission of detection parameters.

3.1.1. Database. Database stores training data set. The training nodes can be classified as normal and abnormal by the tag. The training data set is $D = (V_1^{y=1}, V_2^{y=1}, \ldots, V_{N-1}^{y=-1}, V_N^{y=-1})$ which contains N nodes. y is the tag of the node, $y = 1$ represents normal, and $y = -1$ represents abnormal. Each node $V_i^y = (v_{i1}, v_{i2}, \ldots, v_{im})$ has m-dimensional features. Each feature v_{ij} is normalized as

$$v_{ij}' = \frac{v_{ij} - \min(v_j)}{\max(v_j) - \min(v_j)}. \tag{1}$$

Normalized training data will be utilized to compute the flow detection parameters in EMSOM Preprocessor and KD-tree Builder.

3.1.2. EMSOM Preprocessor. EMSOM Preprocessor computes SOM map search space, which is the number range of SOM neurons, to reduce the computational complexity of searching SOM map. Then, it finds out the appreciated SOM map for the EMSOM-KD detection method and classifies the neurons in the map by entropy measuring.

(1) Calculation of the SOM Map Search Space. SOM neurons represent classification kinds. When the number of neurons is too small, the classification accuracy of SOM may be too low. Nevertheless, a vast number of SOM neurons may cause dead neurons and increase computation complexity. Thus a reasonable range of the SOM neuron number can improve the efficiency and precision of the SOM classifier. SOM is an unsupervised clustering method that can assign each node V to the nearest cluster C_i with a corresponding centroid U_i. The number of neurons set in the SOM map is related to the ideal clustering number of training data. However, the ideal clustering number is often difficult to define, and it is analyzed in detail in research [34]. We estimate the range of the ideal clustering number based on clustering compactness changes. K-means++ is used to compute the search space. Because K-means++ is an efficient unsupervised clustering algorithm that can calculate cluster centroids and is conducive to clustering compactness computation, it is detailed in research [35].

Definition 1. SSE_k is the sum distance of each node to its nearest cluster centroid. It represents the clustering compactness of clustering number k. As k raises, SSE_k can be smaller, and the cluster is more compact. SSE_k is calculated as

$$\mathrm{SSE}_k = \sum_{i=1}^{k} \sum_{V \in C_i} \|V - U_i\|. \tag{2}$$

Definition 2. If SSE_α has the largest relative decrease, α is close to the actual number of data categories according to Elbow Method [36]. α is the lower limit of the ideal clustering number, and it is calculated as (3), where k_m is the maximum value of k.

$$\alpha = \arg\max_{k=2,\ldots,k_m} \left(\frac{\mathrm{SSE}_{k-1} - \mathrm{SSE}_k}{\mathrm{SSE}_k} \right). \tag{3}$$

Definition 3. β is the stable clustering number, $\beta > \alpha$. When k is bigger than β, clustering compactness changes slightly. β is the upper limit of the ideal clustering number. It is calculated as

$$\beta = \arg\max_{k=2,\ldots,k_m} \frac{\left|\mathrm{SSE}_{k-1} - \mathrm{SSE}_k\right|}{\left|\mathrm{SSE}_k - \mathrm{SSE}_{k+1}\right|}. \tag{4}$$

Therefore, the range of the ideal clustering number is $[\alpha, \beta]$. In order to ensure adequate search space for the suitable SOM map, the search space of neuron number is $[\alpha, \varepsilon \cdot \beta]$, and ε is a positive integer. ε grows to expand the search space until the ideal SOM map is found out.

(2) Determination of the Suitable SOM Map. SOM map is two-dimensional [37]; the map size is $L \times R$, where L is the column number and R is the row number. $\mathrm{num}_{L \times R}$ represents the neuron number, which should be in $[\alpha, \varepsilon \cdot \beta]$ during the process of determining a suitable SOM map. The

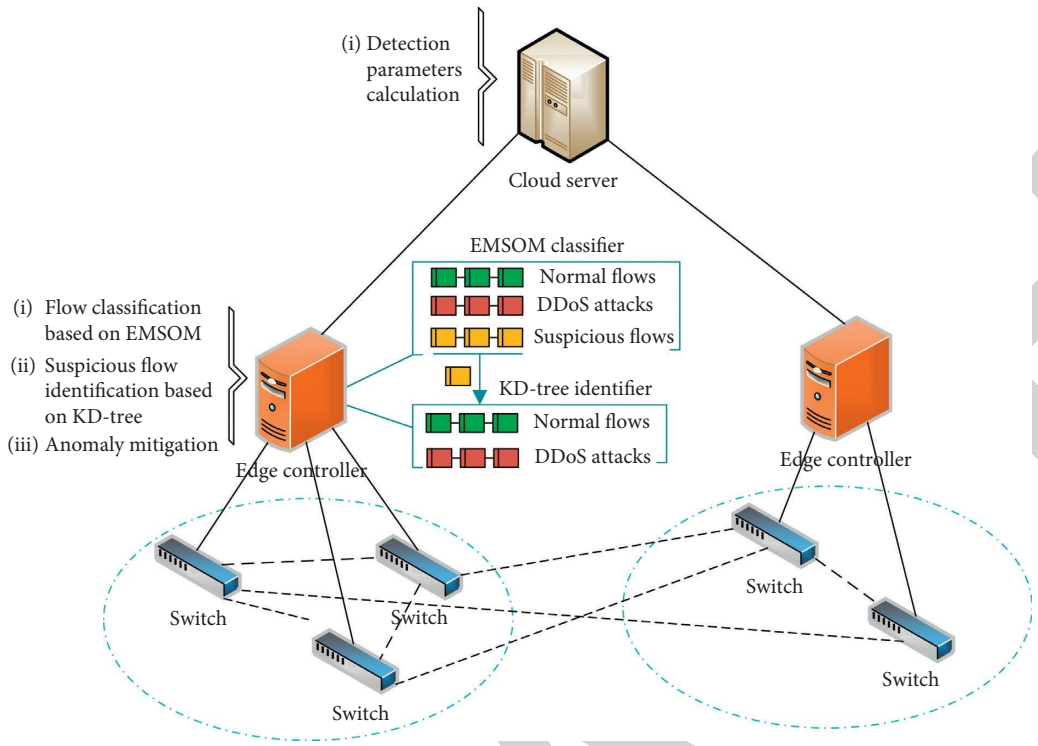

Figure 1: Cloud-edge collaboration architecture for DDoS detection in SDN.

entropy method is used to measure the properties of SOM neurons and score the map. Therefore, EMSOM makes up for the blindness of SOM selection.

Classical SOM map is built and trained by training data, then its neurons are recognized by the statics of various training data. However, there may be dead neurons or suspicious neurons. These neurons can reduce the precision of SOM. Thus neurons are measured and divided into normal, abnormal, and suspicious categories by entropy.

Definition 4. ENT_i is the mapping entropy of the ith neuron in the SOM map. a_i and b_i are the numbers of normal and abnormal training nodes mapped by the ith neuron. ENT_i is computed as

$$\mathrm{ENT}_i = -\left(\frac{a_i}{a_i + b_i}\right)\ln\left(\frac{a_i}{a_i + b_i}\right) - \left(\frac{b_i}{a_i + b_i}\right)\ln\left(\frac{b_i}{a_i + b_i}\right). \tag{5}$$

The greater the mapping information entropy, the more uncertain the neuron. If $a_i = b_i = 0$, ith neuron is a dead neuron that cannot identify the flows; let $\mathrm{ENT}_i = 1$.

$\mathrm{ENT}_i = 0$ means that the ith neuron maps only one kind of training data. If $a_i < b_i$ and $\mathrm{ENT}_i = 0$, the ith neuron can be judged as abnormal; it will be put in abnormal neuron set AN. If $a_i > b_i$ and $\mathrm{ENT}_i = 0$, the ith neuron can be judged as normal and will be put in normal neuron set NN.

$\mathrm{ENT}_i \neq 0$ means that the ith neuron maps both kinds of training flows, and the mapping entropy needs to be compared with the judgment threshold to determine the type of ith neuron.

Definition 5. T is the judgment threshold. It is computed as

$$T = \sum_{i=1}^{L\times R} \frac{\mathrm{ENT}_i}{L \times R}. \tag{6}$$

If $a_i < b_i$ and $\mathrm{ENT}_i \leq T$, the ith neuron is abnormal; it can be put in AN. If $a_i < b_i$ and $\mathrm{ENT}_i > T$, the ith neuron is suspicious and has a strong possibility of misjudging; it will be put in suspicious neuron set SN. Likewise, if $a_i > b_i$ and $\mathrm{ENT}_i \leq T$, ith neuron will be put in NN. Otherwise, it will be placed in SN.

After recognizing neurons in the EMSOM map, the EMSOM map has to be evaluated for its performance.

Definition 6. $\mathrm{SF}_{L\times R}$ is the score of the SOM map performance of filtering out suspicious flows. It is calculated as

$$\mathrm{SF}_{L\times R} = \sum_{i\in SN} \frac{a_i + b_i}{2n}, \tag{7}$$

$\mathrm{SF}_{L\times R}$ shows the ratio of the nodes mapped by suspicious neurons to the total training nodes. The larger $\mathrm{SF}_{R\times L}$ is, the more suspicious traffic the SOM map may filter out.

Definition 7. $\mathrm{SA}_{L\times R}$ is the score of identification accuracy of the SOM map whose topology is $L \times R$. It is calculated as

$$\mathrm{SA}_{L\times R} = \sum_{i\in \mathrm{NN}} \left(\frac{a_i + b_i}{2n} \cdot \mathrm{ENT}_i\right) + \sum_{i\in \mathrm{AN}} \left(\frac{a_i + b_i}{2n} \cdot \mathrm{ENT}_i\right), \tag{8}$$

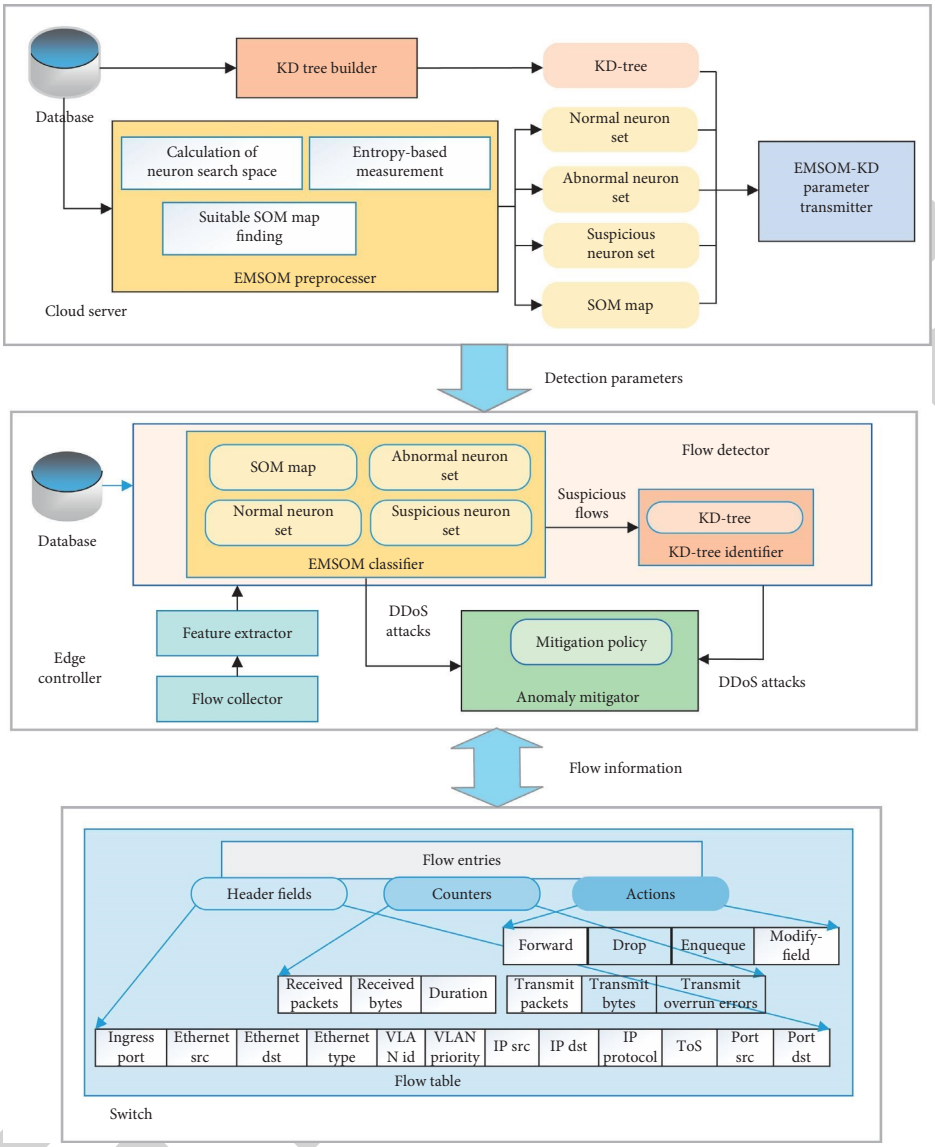

Figure 2: DDoS detection model based on Entropy-Measuring SOM and KD-tree for SDN.

$SA_{L\times R}$ expresses the influence of mapping entropy of normal and abnormal neurons in the SOM map. The larger $SA_{L\times R}$, the lower the identification accuracy of the SOM map.

The SOM map deployed in the controller should filter suspicious traffic as little as possible whereas accurately distinguishing regular traffic and attack traffic. The score expression of the SOM map is

$$\text{Score}_{L\times R} = e^{\text{SF}_{L\times R}} + e^{\sqrt{\text{SA}_{L\times R}}}. \tag{9}$$

According to the rule of $\text{SF}_{L\times R}$ and $\text{SA}_{L\times R}$, $\text{Score}_{L\times R}$ has the nature that the lower the SOM map score, the better the performance. A suitable SOM map can be expressed by the formula

$$\text{estmap}(L\times R) = \underset{\text{num}_{L\times R}\in[\alpha,\varepsilon\cdot\beta]}{\arg\min}\ \text{Score}_{L\times R}. \tag{10}$$

Figure 3 shows the computation process of the suitable SOM map. The detailed steps of neuron classification in the SOM map and determination of the best map for EMSOM-KD are as follows:

① SOM topology creation: in the map search space of $[\alpha, \varepsilon\cdot\beta]$, create the SOM topology $L\times R$, whose neuron number is$\text{num}_{L\times R}$ and $\alpha \le \text{num}_{L\times R} \le \varepsilon\cdot\beta$.

② Network initialization: create $S((\alpha \le S \le \varepsilon\cdot\beta)$ neurons $W_1, W_2, \ldots, W_S$. Each neuron has m-dimensional weights $W_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, $1 \le i \le S$; the weights are initialized by random values.

③ Winning neuron acquisition: input the training vector V_h, and calculate the distances between the vector and neurons as

$$\text{Dis}(V_h, W_i) = \|V_h - W_i\| = \sqrt{\sum_{j=1}^{m}(v_{hj} - w_{ij})^2}. \tag{11}$$

The neuron with the smallest distance is selected to be the winning neuron.

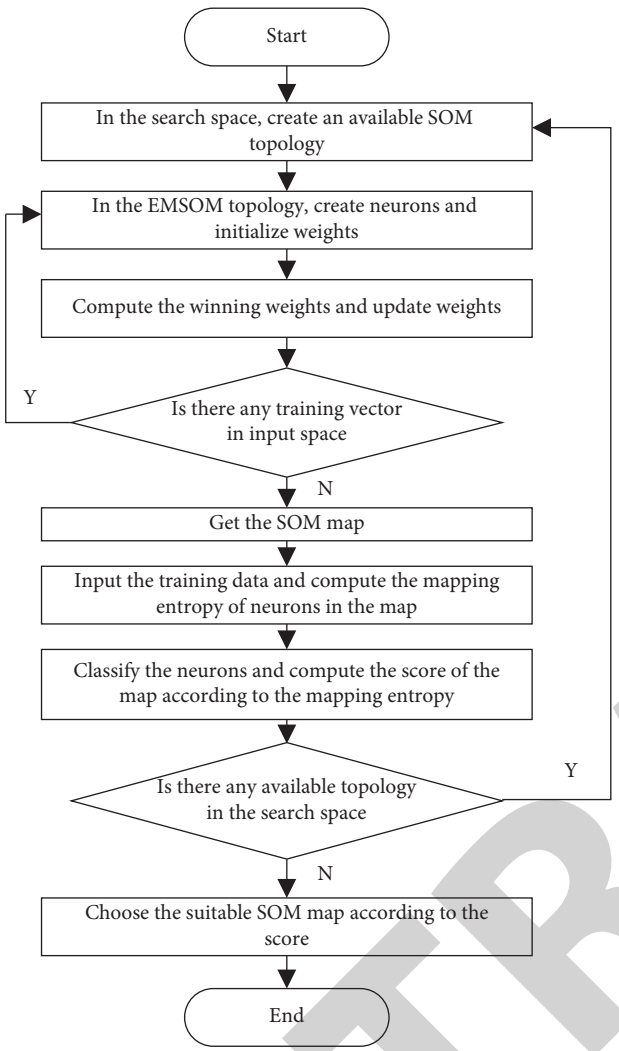

Figure 3: Determination process of the suitable SOM map.

④ Weights update: collect the neighboring neurons of the winning neuron W_z, and update the weights of W_z and its neighbors as

$$W_z(t+1) = W_z(t) + \eta(t)\alpha(t)\left(V_h(t) - W_z(t)\right), \tag{12}$$

$\eta(t)$ is the neighborhood function, and $\alpha(t)$ is the learning rate.

⑤ Loop: repeat steps 2 to 3, until there is no more training vectors in input space, and get the trained SOM map Δ.

⑥ Entropy measurement of neurons: input training data into the SOM map Δ, and compute the best match neuron for each training node by (13).

$$l_i = \arg\min_{z\in\Delta}\left(\left\|V_i - W_z\right\|\right). \tag{13}$$

Then, count each neuron number in each training category and calculate each neuron's mapping entropy by formula (5).

⑦ Classification of neurons: compute the judgment threshold as formula (6). Then assign the neurons into normal neuron set NN, abnormal neuron set AN, and suspicious neuron set SN due to the mapping entropies and the judgment threshold.

⑧ Score computation of SOM map Δ: calculate the score of the SOM map Δ to evaluate the performance of Δ by the formulas (7)–(9).

⑨ Suitable SOM map selection: repeat steps 1 to 8, until there is no more available EMSOM topology in the map search space, then choose the suitable SOM map by formula (10).

3.1.3. KD-Tree Builder. KD-tree is an improvement of KNN. It can quickly find the nearest training points to the target node through the tree structure index without calculating the distance between the target node and each data in the training set.

KD-tree Builder constructs a balanced binary tree through a recursive method to store training data. Due to the number of training data set features, the binary tree divides an entire feature space into specific parts for fast query operations. The constructed KD-tree will be transmitted to the controller for the inspection of suspicious flows. Details of KD-tree construction are explained in research [38].

3.1.4. EMSOM-KD Parameter Transmitter. Before traffic detection, each controller needs to be registered on the cloud server, and the cloud will verify the controller identity. After verification, the controller sends a parameter request to the cloud server. EMSOM-KD Parameter Transmitter then sends training data, SOM map, normal neuron set, abnormal neuron set, suspicious neuron set, and KD-tree to the controller.

3.2. Detection Modules in Edge Controller. After receiving the parameters from the cloud server, the edge controller can efficiently detect network traffic. Detection modules in the controller include Flow Collector, Feature Extractor, Flow Detector, and Anomaly Mitigator.

3.2.1. Flow Collector. Flow Collector regularly communicates with switches and collects the flow information, which contains IP protocol, IP source/destination address, source/destination port, the numbers of received packets, received bytes, duration, etc. The flow information is helpful for the identification of attack traffic. And, it will be transported to Feature Extractor for feature computation.

3.2.2. Feature Extractor. This module extracts feature vectors from the collected flow information. Network flows can be classified through the flow feature vectors whose elements are interconnected and reflect network condition characteristics.

During the DDoS process, the attacker may use different protocols to attack the specific destination ports. For

example, HTTP flooding mainly occupies port 80. Thus, protocol and destination port are related to DDoS attacks. Moreover, the rate and size of the flow also reflect the law and characteristics of attacks. For example, low-rate DDoS periodically launches malicious attack traffic at a low rate, and the packet size of network flow may change regularly. Therefore, it is necessary to count the flow duration and calculate the average packet size APS in each flow by

$$\mathrm{APS} = \frac{\sum_{i=1}^{\mathrm{FE}_j\_\mathrm{Packetnum}} \mathrm{Packet\_size}_i}{\mathrm{FE}_j\_\mathrm{Packetnum}}, \tag{14}$$

FE_j_Packetnum is the packet number of the flow entry. $\mathrm{Packet\_size}_i$ is the length of ith packet of the jth flow. APS can describe the flow size.

During the DDoS attacking process, multiple sources are used to send massive data to the victim server, which will become unavailable for legal users. Thus, DDoS attacks can increase traffic sharply, so traffic generating speed reflects the network condition. PR is the flow packets rate that is the number of packets transferred per second. BR is the flow byte rate that is the number of packets transmitted per second. PR and BR are calculated by

$$\mathrm{PR} = \frac{\mathrm{FE}_j\_\mathrm{Packetnum}}{\mathrm{duration}}, \tag{15}$$

$$\mathrm{BR} = \frac{\mathrm{FE}_j\_\mathrm{Bytenum}}{\mathrm{duration}}, \tag{16}$$

FE_j_Bytenum is the byte number of the flow. Therefore, the feature vector comprises protocol, flow duration, destination Port, APS, PR, and BR.

3.2.3. Flow Detector. In Flow Detector, EMSOM Classifier, and KD-tree Identifier work together to detect network flows. Figure 4 illustrates the flow detection process, which contains two stages: flow classification and suspicious traffic filtering based on EMSOM and suspicious flow identification based on KD-tree.

EMSOM Classifier calculates the best match neuron for the network flow in the first stage and divides them into normal, malicious, and suspicious according to the types of neurons. And suspicious flows should be transported to KD-tree Identifier for fine-grained recognition. Because EMSOM Preprocessor picks out dead neurons and suspicious neurons that may lead to a great classification error rate, the accuracy of EMSOM can be improved. In the second stage, KD-tree Identifier uses Best Bin First (BBF) [22] algorithm to search the g nearest training nodes of the suspicious flow in the KD-tree and computes the node number in each category. If most of the closest training nodes are normal, then the suspicious flow is identified as normal; otherwise, the suspicious flow is judged as a DDoS attack. The EMSOM-KD detection method is described as Algorithm 1.

3.2.4. Anomaly Mitigator. When Flow Detector finds DDoS attacks, it sends the attacking flow information to Anomaly Mitigator. Anomaly Mitigator modifies the action field in the flow table and sends modified flow tables to the OpenFlow switch to discard attacking flows. What is more, Anomaly Mitigator sends information about the attack flows (such as MAC, IP, port) and defense instructions to the firewall.

4. Experiments and Performance Evaluation

This section introduces the testing environment and process parameter adjustment and presents experiment details. The experiment results are analyzed for performance evaluation of EMSOM-KD.

4.1. Testing Environment. Figure 5 presents the experimental topology, which includes a Ryu controller, the cloud server, OpenFlow switches, legal user hosts, and attacking hosts. Before flow detection, the cloud server deploys the training data set and preprocessed data in the controller. Legitimate hosts use network applications to generate regular traffic. The attacks use a DDoS tool such as Kali to develop DDoS attacks. The training data set has 4000 flows, including 2000 normal flows and 2000 abnormal flows. The initial EMSOM-KD algorithm parameters are shown in Table 1.

We use recall of attacking flows R_a, precision of attacking flows P_a, and $F1$ score to evaluate the performance of EMSOM-KD. $F1$ can measure the accuracy of the detection method. The larger $F1$, the higher the accuracy of the method. R_a, P_a, and $F1$ are calculated as (17)–(19).

$$R_a = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{17}$$

$$P_a = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{18}$$

$$F1 = \frac{2R_aP_a}{R_a + P_a}, \tag{19}$$

TP is the number of the attacking flows that are identified correctly. FN is the number of the attacking flows that are misjudged. FP is the number of the normal flows that are mistaken.

4.2. Parameter Adjustment in Cloud Server. The cloud server selects the suitable SOM map and classifies neurons in the map using the entropy measuring method in Section 3. Then, the parameters will be deployed in the edge SDN controller for DDoS detection.

In order to find a suitable SOM map, we use the K-means++ algorithm to cluster training nodes and calculate SSE_k, $(\mathrm{SSE}_{k-1} - \mathrm{SSE}_k)/\mathrm{SSE}_k$ and $(|\mathrm{SSE}_{k-1} - \mathrm{SSE}_k|/|\mathrm{SSE}_k - \mathrm{SSE}_{k+1}|)$ for the different numbers of clusters. The range of cluster number k is set as [2, 100]. The calculation results are shown in Table 2.

When $k = 7$, $(\mathrm{SSE}_{k-1} - \mathrm{SSE}_k)/\mathrm{SSE}_k$ has the max value, and $k = 38$, $(|\mathrm{SSE}_{k-1} - \mathrm{SSE}_k|/|\mathrm{SSE}_k - \mathrm{SSE}_{k+1}|)$ reaches a maximum. Thus, $\alpha = 7$ and $\beta = 38$. As shown in Figure 6, α is the knee point, and SSE_k is stable after β. Let $\varepsilon = 2$, and the search space of the neuron number is [7, 76].

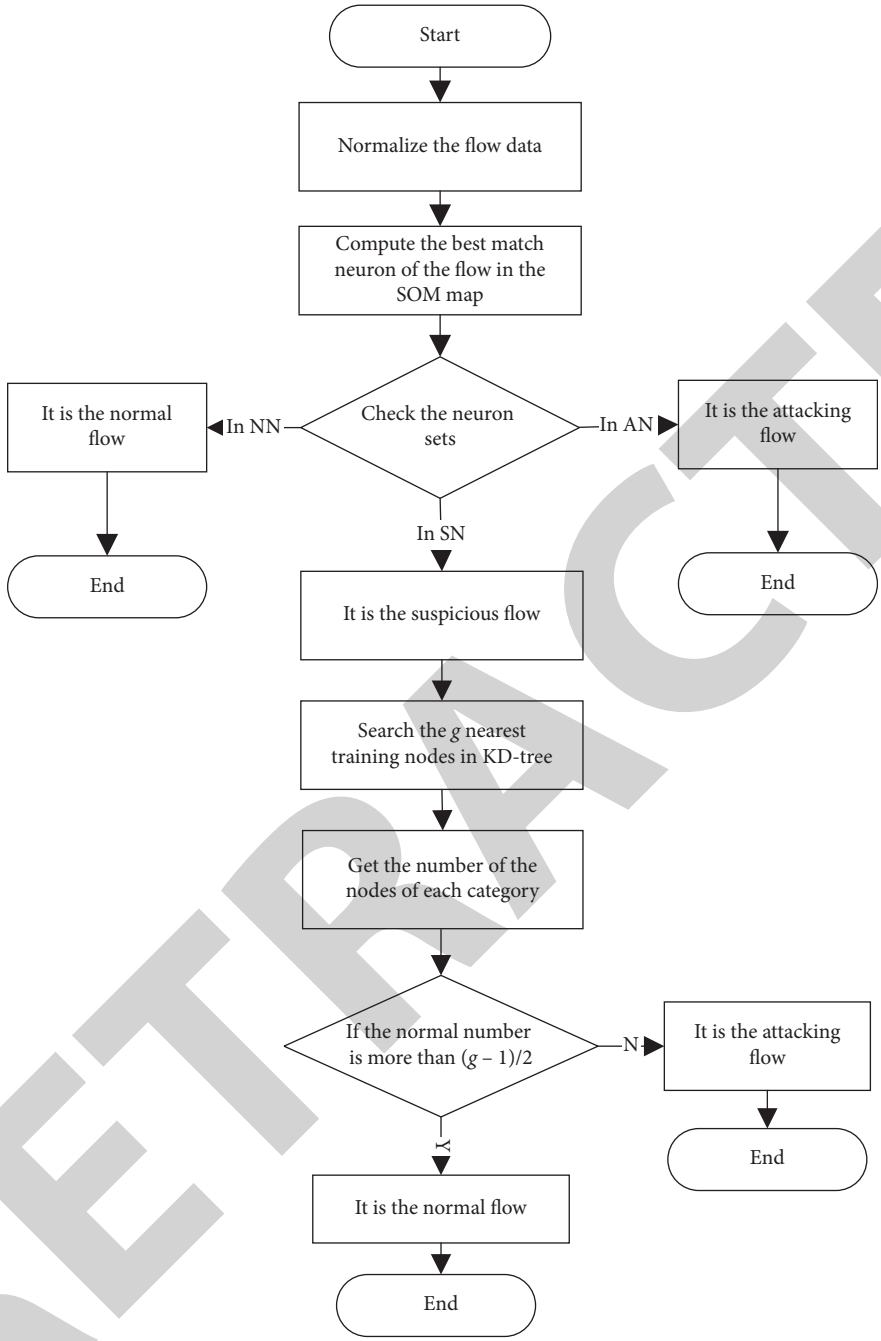

Figure 4: Flow detection process based on EMSOM-KD.

Input: the detected flow vector, SOM map, abnormal neuron set AN, normal neuron set NN, suspicious neuron set SN, KD-tree.
Output: the detection result.
(1) For each network flow
(2) Normalize the detected flow vector by (1).
(3) Compute the best match neuron in the suitable SOM map.
(4) If the best match neuron is in NN, then
The detected flow is normal.
Else if the best match neuron is in AN, then
The detected flow is abnormal.

Algorithm 1: Continued.

```
        Else
          The detected flow is suspicious.
        End if
(5) End for
(4) For each suspicious flow
        Search the g nearest nodes in the KD-tree.
        Count the number of nodes of each type.
        If the number of normal nodes is more than (g − 1)/2, then
          The detected flow is normal.
        Else
          The detected flow is abnormal.
    End for
```

Algorithm 1: DDoS detection based on EMSOM-KD.

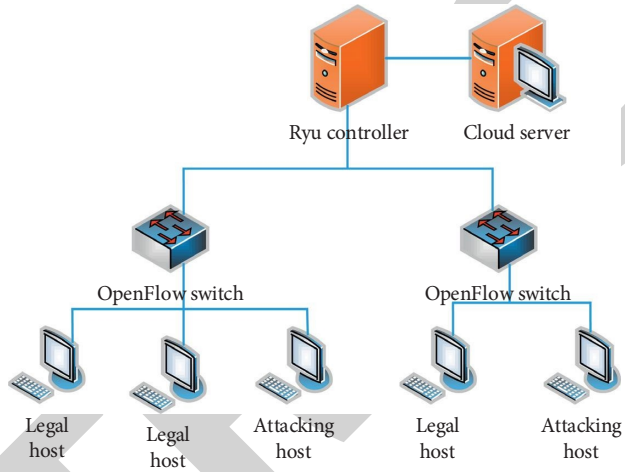

Figure 5: Experimental implementation topology.

Table 1: Values of EMSOM-KD parameters.

SOM parameter	Value
Number of training epoch	100
Order learning rate	0.9
Tuning learning rate	0.02
Number of nearest nodes of KD-tree	7

Table 2: SSE_k, $(SSE_{k-1} - SSE_k)/SSE_k$, and $(|SSE_{k-1} - SSE_k|/|SSE_k - SSE_{k+1}|)$ of the different cluster numbers.

k	SSE_k	$(SSE_{k-1} - SSE_k)/SSE_k$	$(\|SSE_{k-1} - SSE_k\|/\|SSE_k - SSE_{k+1}\|)$
5	719.5819	0.5426	3.4598
6	832.4315	−0.1356	0.2548
7	389.5981	1.1366	12.2015
8	353.3047	0.1027	0.7687
9	306.0899	0.1543	0.8909
36	101.5119	0.0695	0.3009
37	124.9472	−0.1876	1.0414
38	102.4431	0.2197	19.0177
39	101.2598	0.0117	1.1833

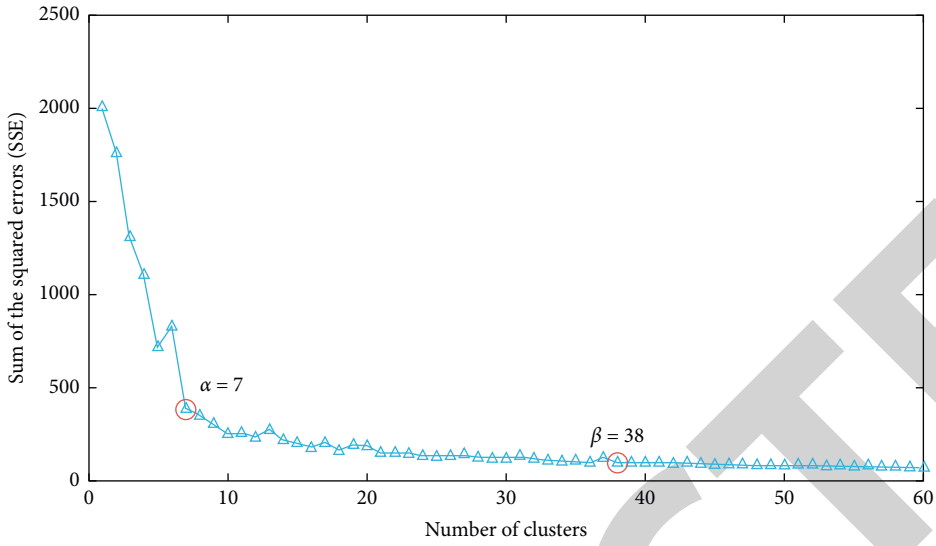

Figure 6: SSE_k values with different k.

Table 3: Performance and score of different SOM maps.

Map size $L \times R$	$SF_{L\times R}$	$SA_{L\times R}$	$Score_{L\times R}$	Suspicious flow number	$F1$
4×11	0.1585	0.0299	2.3605	786	0.9944
5×11	0.1403	0.0314	2.3446	698	0.9929
6×11	0.146	0.0246	2.3269	1145	0.9940
4×12	0.2205	0.0133	2.3694	1541	0.9958
5×12	0.1713	0.0100	2.2918	837	0.9977
6×12	0.1508	0.0095	2.2652	751	0.9964
4×13	0.1293	0.0177	2.2802	1072	0.9939
5×13	0.1573	0.0079	2.2633	781	0.9977
7×10	0.141	0.0151	2.2822	711	0.9954
7×9	0.1533	0.0118	2.2802	1188	0.9971

We utilize the scoring method to find a suitable SOM map. Five thousand test flows are used to evaluate the performance of each SOM map in the search space. As shown in Table 3, the smaller the score, the greater the possibility that the map can detect most flows and has high detection accuracy. We choose $L = 5, R = 13$ as the suitable SOM map.

The neurons in this suitable map are divided into normal, abnormal, and suspicious by the entropy measurement. The neuron classification result of the suitable SOM map is shown in Figure 7.

4.3. Performance Evaluation of EMSOM-KD. The proposed method is tested with 2000 to 20000 flows, containing the same number of DDoS attacking flows and normal flows. What is more, we compare EMSOM-KD with SOM type algorithms such as SOM [29] and DSOM [30], and fast KNN type algorithms such as KD-tree [25], SOM-KD [31]. SOM-KD replaces the original training set with the trained neurons to calculate the nearest neighbor nodes, so it belongs to the KNN type. SOM and EMSOM-KD have the same map size. DSOM map size is 10×15, and SOM-KD map size is 20×15.

Figure 8 illustrates the ratio of suspicious flows filtered through the suitable SOM map to total flows. The radio of suspicious flows is less than 16%. It means that EMSOM can directly identify most attacking and normal flows and filter out a small number of suspicious flows that EMSOM cannot determine. Some normal flows are similar to DDoS attacks, so the suspicious flows include DDoS attacks and normal ones.

There are suspicious and dead neurons in the traditional SOM map, which affects the detection accuracy of SOM. EMSOM takes advantage of entropy measurement to exclude suspicious neurons and dead neurons and determine the suitable SOM map for high-precision flow identification. As shown in Figure 9, $F1$ value of EMSOM evaluating the direct classification of normal and abnormal flows is higher than 0.995. $F1$ of KD-tree assessing the identification of suspicious flows filtered by EMSOM is more than 0.965. Because suspicious flows have a small amount, the accuracy of EMSOM-KD is still higher than 0.99.

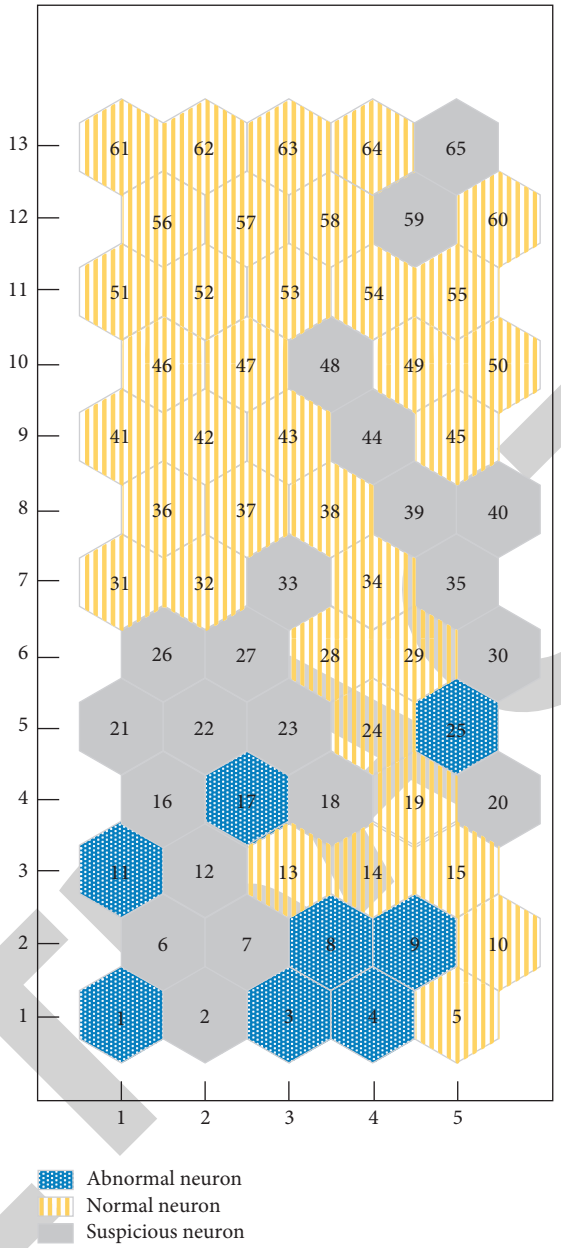

Figure 7: Neuron classification in the suitable SOM map.

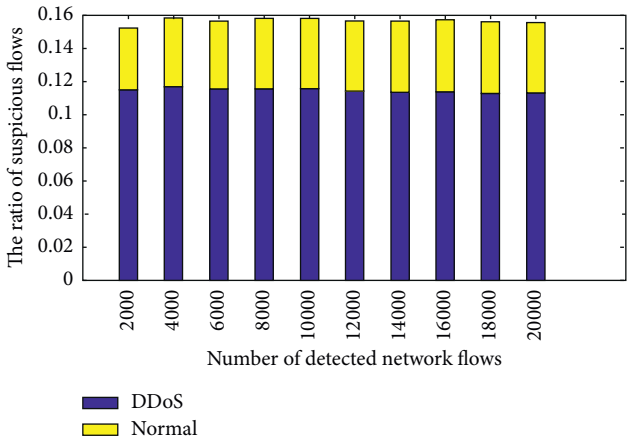

Figure 8: The radio of suspicious flows filtered by EMSOM.

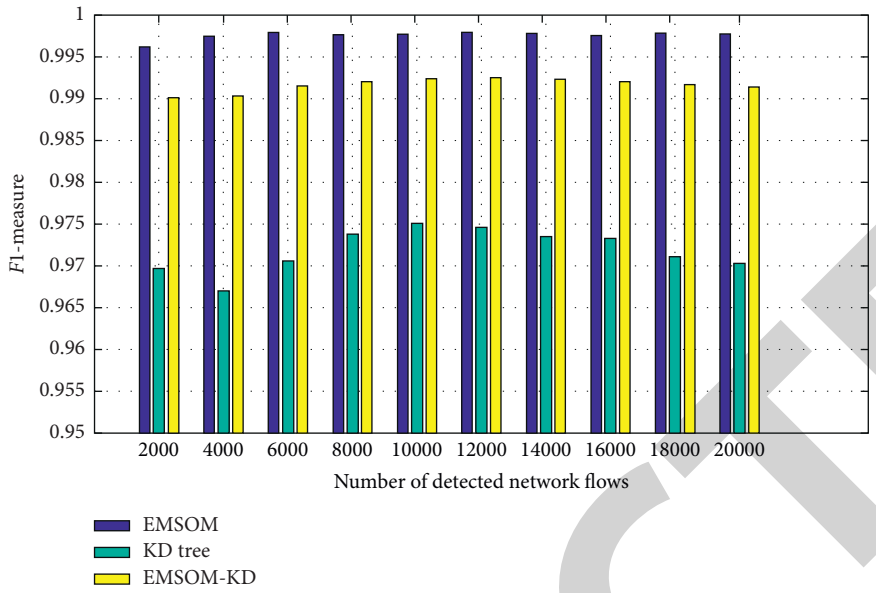

Figure 9: The detection accuracy of EMSOM-KD.

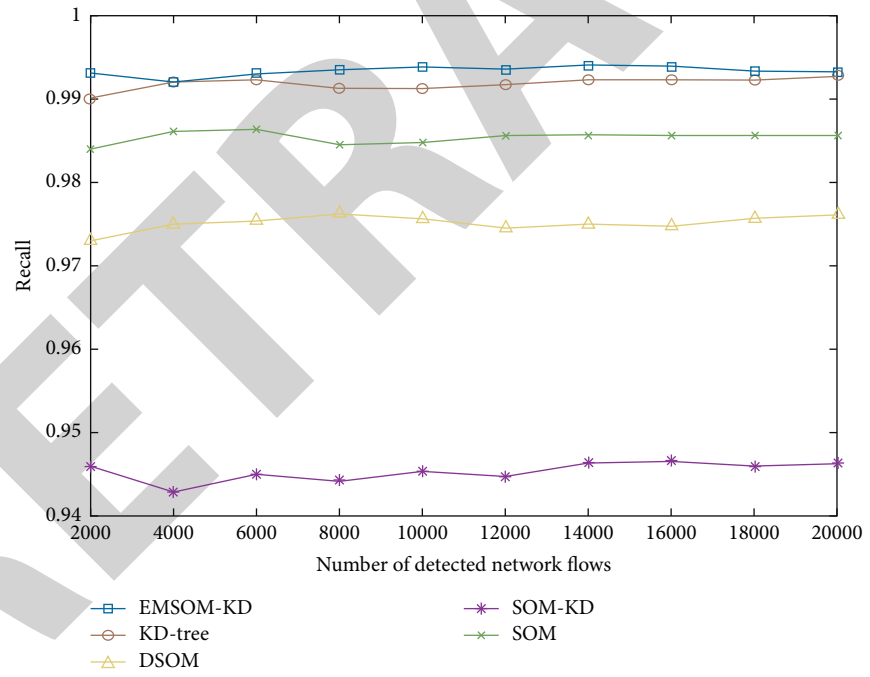

Figure 10: Recall of EMSOM-KD and other detection methods.

As shown in Figures 10 and 11, both recall and precision of EMSOM-KD are better than other detection methods. That is, EMSOM-KD has the lowest error rates of normal traffic and DDoS attack recognition. It implies that using EMSOM-KD for DDoS mitigation is conducive to maintain regular network communication in SDN. Figure 12 illustrates that, compared with other algorithms, EMSOM-KD has the best $F1$ score. Therefore, the proposed DDoS detection method has the highest detection accuracy.

Figure 13 shows the detection time of different methods. As the number of flows grows, the detection time of all methods will increase. The consuming time of EMSOM-KD

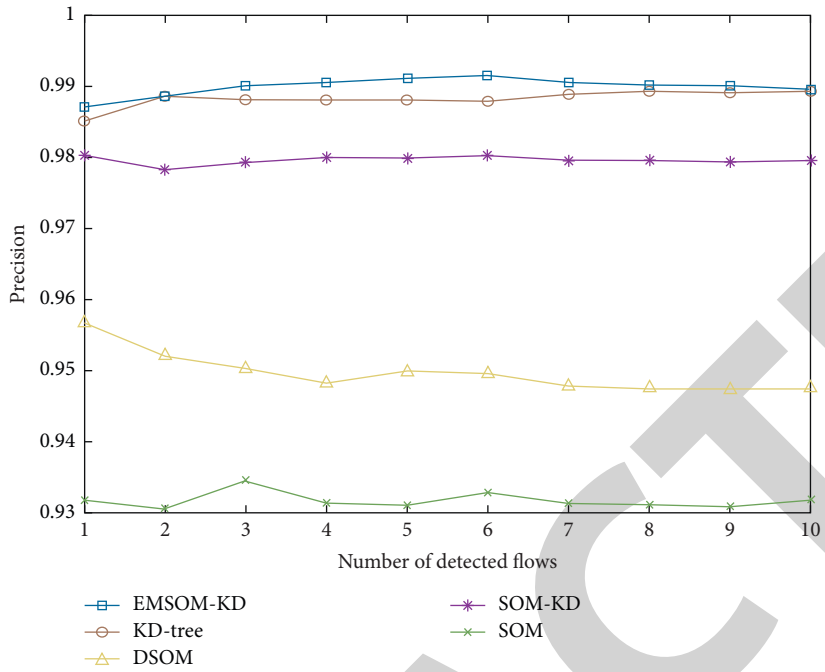

Figure 11: Precision of EMSOM-KD and other detection methods.

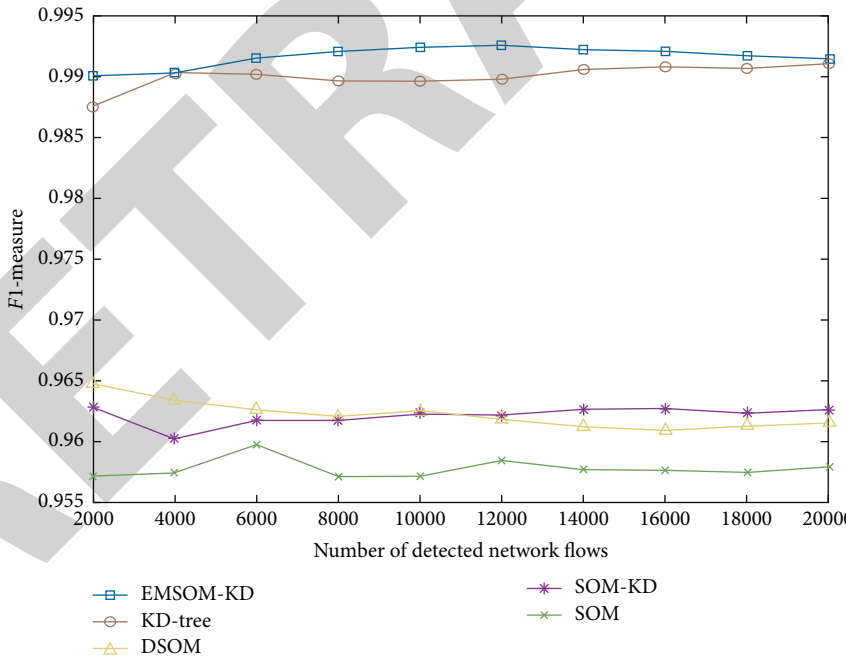

Figure 12: $F1$-measure of EMSOM-KD and other detection methods.

is larger than SOM type methods but is much shorter than KNN type methods.

During the EMSOM-KD detection process, KD-tree needs to identify suspicious traffic additionally. It increases the detection time of EMSOM-KD compared with SOM type methods. As depicted in Figure 14, KD-tree takes up most of the inspection time during the detection process of EMSOM-KD. In other words, the less suspicious flows, the more efficient EMSOM-KD. And the amount of suspicious traffic is small, which reduces the consuming time of KD-tree.

In conclusion, EMSOM-KD improves the detection accuracy of SOM and KD-tree. Moreover, EMSOM-KD takes advantage of SOM to obtain better detection efficiency compared with KD-tree.

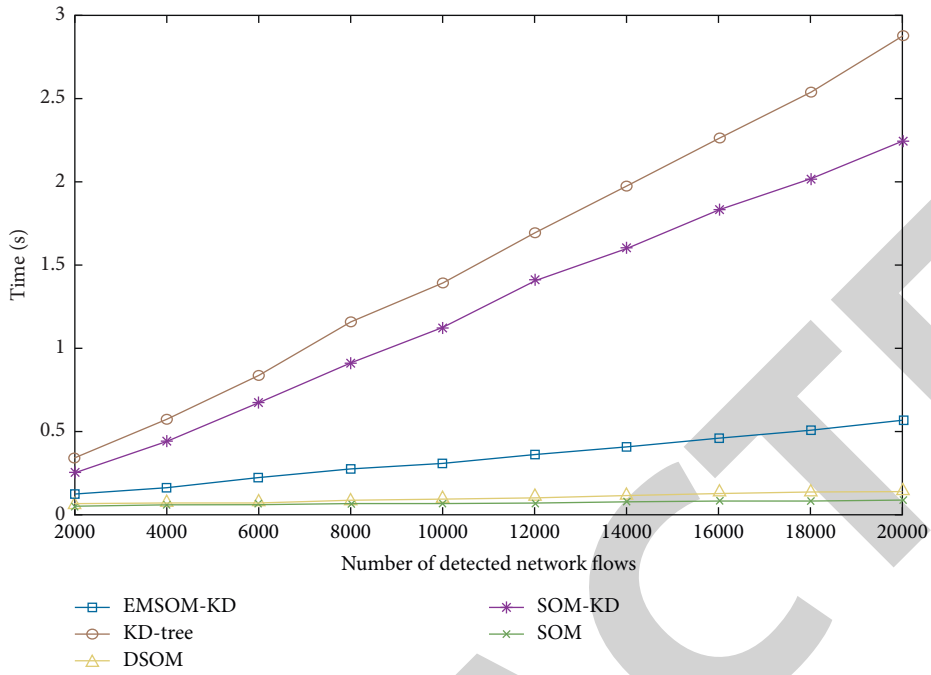

Figure 13: Detection time of different methods.

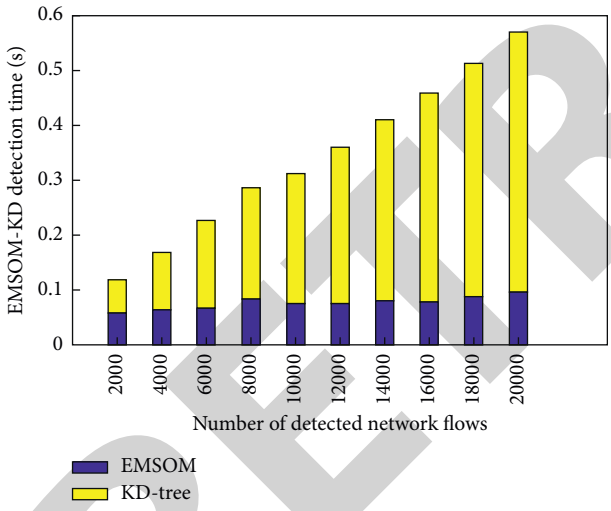

Figure 14: Detection time of EMSOM.

5. Conclusion and Future Work

SDN improves network flexibility and programmability through centralized control. However, it is vulnerable to DDoS network attacks, which leads to network paralysis. Therefore, it is important to protect network security against DDoS in SDN. In this paper, a cloud-edge collaboration detection system is designed for efficient and precise DDoS detection, and a flow detection method based on EMSOM-KD is proposed. EMSOM overcomes the blindness of SOM map selection through the entropy measurement method. It divides flows into three categories: normal, abnormal, and suspicious. Then KD-tree performs fine-grained identification of doubtable flows. Moreover, we did detailed experiments for EMSOM-KD. The experimental results verified the efficiency and accuracy of the proposed method.

Although this article proposes a Cloud-Edge Collaboration Method for DDoS detection in SDN, it is assumed that there is secure communication between the controller and the cloud server. However, if the controller and the cloud server are not in a secure communication environment, and the parameters may be tampered with, the controller cannot perform DDoS detection. In the future, we will study the signature encryption technology for secure communication between the cloud server and the controller. The cloud server will sign and encrypt the parameters. After receiving the parameters, the controller will verify the integrity and validity of the data by decryption.

Moreover, EMSOM-KD can improve the accuracy of SOM and KD-tree. Still, it depends on the historical training data. Our method will be enhanced by automatically collecting more training flows and updating parameters of EMSOM-KD for further DDoS inspection accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest for this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61672299,

61972208, and 61802200; Natural Science Foundation of Jiangsu Province under Grant no. BK20180745; and Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant no. KYCX19_0914.

References

[1] G. Kaur and P. Gupta, "Classifier for DDoS attack detection in software defined networks," *Internet of Things in Business Transformation: Developing an Engineering and Business Strategy for Industry 5.0*, vol. 20, pp. 71–90, 2021.

[2] S. A. Gagangeet, C. Rajat, K. Kuljeet et al., "SAFE: SDN-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Transactions On Industrial Informatics*, vol. 15, no. 1, pp. 469–480, 2019.

[3] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706–5712, 2020.

[4] R. Muñoz, R. Vilalta, R. Casellas et al., "Orchestration of optical networks and cloud/edge computing for IoT services," 2019.

[5] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," in *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia*, pp. 1–6, Association for Computing Machinery, Moscow, Russia, 2013.

[6] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, 2020.

[7] M. P. Singh and A. Bhandari, "New-flow based DDoS attacks in SDN: taxonomy, rationales, and research challenges," *Computer Communications*, vol. 154, pp. 509–527, 2020.

[8] N. Dao, T. V. Phan, J. Kim, T. Bauschert, and S. Cho, "Securing heterogeneous Iot with intelligent DDOS attack behavior learning," 2017.

[9] K. Johnson Singh and T. De, "Mathematical modelling of DDoS attack and detection using correlation," *Journal of Cyber Security Technology*, vol. 1, no. 3-4, pp. 175–186, 2017.

[10] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," 2018.

[11] T. V. Phan, T. G. Nguyen, N.-N. Dao, T. T. Huong, N. H. Thanh, and T. Bauschert, "DeepGuard: efficient anomaly detection in SDN with fine-grained traffic flow monitoring," *IEEE Transactions On Network and Service Management*, vol. 17, no. 3, pp. 1349–1362, 2020.

[12] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang et al., "An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers," *Technologies*, vol. 9, no. 1, 2021.

[13] J. Galeano-Brajones, J. Carmona-Murillo, J. F. Valenzuela-Valdés et al., "Detection and mitigation of dos and ddos attacks in iot-based stateful sdn: an experimental approach," *Sensors*, vol. 20, no. 3, 2020.

[14] R. F. Fouladi, O. Ermiş, and E. Anarim, "A DDoS attack detection and defense scheme using time-series analysis for SDN," *Journal of Information Security and Applications*, vol. 54, 2020.

[15] N. Z. Bawany and J. A. Shamsi, "SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks," *Journal of Network and Computer Applications*, vol. 145, 2019.

[16] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," *Future Generation Computer Systems*, vol. 89, pp. 685–697, 2018.

[17] A. B. Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *The Journal of Supercomputing*, vol. 34, pp. 1–33, 2020.

[18] R. Li and B. Wu, "Early detection of DDoS based on φ-entropy in SDN networks," 2020.

[19] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," *Future Generation Computer Systems*, vol. 97, pp. 275–283, 2019.

[20] M. Shakil, A.F. Y. Mohammed, R. Arul et al., *A Novel Dynamic Framework to Detect DDoS in SDN Using Metaheuristic Clustering*, Transactions on Emerging Telecommunications Technologies, Shanghai, China, 2019.

[21] Y. Chen, J. Pei, and D. Li, "DETPro: a high-efficiency and low-latency system against DDoS attacks in SDN based on decision tree," 2019.

[22] M. Latah and L. Toker, "Towards an efficient anomaly-based intrusion detection for software-defined networks," *Iet Networks*, vol. 7, no. 6, pp. 453–459, 2018.

[23] N. N. Tuan, P. H. Hung, N. D. Nghia et al., "A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN," *Electronics*, vol. 9, no. 3, 2020.

[24] S. Dong and M. Sarem, "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2019.

[25] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE Journal On Selected Areas in Communications*, vol. 36, no. 3, pp. 628–643, 2018.

[26] Z. Liu, Y. He, W. Wang, and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Communications*, vol. 16, no. 7, pp. 144–155, 2019.

[27] O. Hannache and M. C. Batouche, "Neural network-based approach for detection and mitigation of DDoS attacks in SDN environments," *International Journal of Information Security and Privacy*, vol. 14, no. 3, pp. 50–71, 2020.

[28] B. Han, X. Yang, Z. Sun, J. Huang, and J. Su, "OverWatch: A cross-plane DDoS attack defense framework with collaborative intelligence in SDN," *Security and Communication Networks*, vol. 2018, 2018.

[29] T. Wang and H. Chen, "SGuard: A lightweight SDN safeguard architecture for DoS attacks," *China Communications*, vol. 14, no. 6, pp. 113–125, 2017.

[30] T. V. Phan, N. K. Bao, and M. Park, "Distributed-SOM: a novel performance bottleneck handler for large-sized software-defined networks under flooding attacks," *Journal of Network and Computer Applications*, vol. 91, pp. 14–25, 2017.

[31] T. M. Nam, P. H. Phong, T. D. Khoa et al., "Self-organizing map-based approaches in DDoS flooding detection using SDN," 2018.

[32] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: a social multimedia perspective," *IEEE Transactions On Multimedia*, vol. 21, no. 3, pp. 566–578, 2019.

[33] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 493–512, 2013.

[34] A. Gupta, S. Datta, and S. Das, "Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering," *Pattern Recognition Letters*, vol. 116, pp. 72–79, 2018.
[35] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," 2006.
[36] D. Marutho, S. H. Handaka, and E. Wijaya, "The determination of cluster number at *k*-mean using elbow method and purity evaluation on headline news," 2018.
[37] D. Miljković, "Brief review of self-organizing maps," 2017.
[38] P. Ram and K. Sinha, "Revisiting kd-tree for nearest neighbor search," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1378–1388, Association for Computing Machinery, Anchorage, AK, USA, 2019.