An Optimal Packet Assignment Algorithm for Multi-level Network Intrusion Detection Systems

Dao Thi-Nga[1(✉)], Chi Hieu Ta[1], Van Son Vu[1], and Duc Van Le[2]

[1] Le Quy Don Technical University, Hanoi 10000, Vietnam
daothinga.mta@gmail.com, hieunda@gmail.com, sontlc246@gmail.com
[2] Nanyang Technological University, Singapore 639798, Singapore
vdle@ntu.edu.sg

Abstract. With the outbreaks of recent cyber-attacks, a network intrusion detection system (NIDS) which can detect and classify abnormal traffic data has drawn a lot of attention. Although detection time and accuracy are important factors, there is no work considering both contrastive objectives in an NIDS. In order to quickly and accurately respond to network threats, intrusion detection algorithms should be implemented on both fog and cloud devices, which have different levels of computing capacity and detection time, in a collaborative manner. Therefore, this work proposes a packet assignment algorithm that assigns detection and classification tasks for appropriate processing devices. Specifically, we formulate a novel optimization problem that minimizes detection time while achieving accuracy performance and computational constraints. Then, an optimal packet assignment algorithm that allocates as many packets as possible to fog devices in order to shorten the detection time is proposed. The experimental results on a state-of-the-art network dataset (UNSW-NB15) show that the proposed packet assignment algorithm produces similar performance to the optimal solution with regard to the detection time and accuracy.

Keywords: Network intrusion detection · Packet assignment · Internet of things

1 Introduction

The advanced development of communication devices and networking technologies has enabled the explosive growth in data exchange and associated services in computer networks [11], but also introduced extra challenges for network security due to the increase of emerging cyber-attacks. A recent report by Forbes has shown that the measured traffic of network attacks increases by three-fold to more than 2.9 billion events worldwide in 2019 [3]. Therefore, it is essential to develop a robust network intrusion detection system (NIDS) that can quickly detect network intrusions with a high detection accuracy.

N.-S. Vo and V.-P. Hoang (Eds.): INISCOM 2020, LNICST 334, pp. 301–313, 2020.
https://doi.org/10.1007/978-3-030-63083-6_23

A number of previous studies [1,4,5,8,12,15] have proposed various network intrusion detection (NID) schemes which aim at detecting the network intrusions based on advanced machine learning (ML) techniques. For instance, Moustafa et al. [8] proposed an ensemble NID algorithm which combines different ML models, including the decision tree, Naïve Bayes, and neural network to detect abnormal packets generated by the network intrusions.

In addition, the study in [5] proposed a two-phase deep learning model which can detect multiple types of network attacks. Specifically, their model first calculates a probability that the network is attacked by network threats based on the collected network data. Then, the network attack probability and other network features are fed to a multi-label classifier for detecting types of the network attacks.

With the primary focus on obtaining a high intrusion detection accuracy, those previous schemes [1,4,5,8,12,15] are computationally expensive in general. Thus, those schemes may not be implemented on real networking devices (e.g., gateways and switches) that often have limited computing and memory capabilities. In those studies, the collected network traffic data are often forwarded to an external computing node (e.g., a fog computing node or a centralized server) with high computing performance CPU/GPUs and large memory, in which the data are processed to detect the network intrusions. As a result, those schemes generally have long detection latencies due to the long communication delay of transmitting the collected data to the external computing devices.

To achieve low NID latency, few existing studies [6,9] have developed low computational complexity NID algorithms which can be implemented in resource-constrained networking devices (e.g., an FPGA-based gateway). For instance, the authors in [6] developed and implemented an entropy-based NID scheme on the data plane of the programmable network devices using the P4 programming language. More specifically, source and destination IP addresses are collected and processed for detecting the DDoS attacks in real time. By processing the collected data at the networking devices rather than sending the data to the external stations, those studies can significantly reduce the NID latency. However, they often suffer from the low detection accuracy due to the limited computational complexity.

Generally, existing algorithms are suitable for a specific type of processing devices and as a result only one factor (detection time or accuracy) can be achieved. There were some initial works [10,14] on multi-level collaborative NIDS which leverages the use of both the fog and the cloud computing levels. However, there is no study which considers both detection time and accuracy factors. In order to address the limitation of existing works, we propose a packet assignment algorithm which can balance between two contrastive requirements (i.e., low detection time and high accuracy). In the network model, we assume that two types of processing devices participate in the intrusion detection tasks in a collaborative manner. Since gateways can detect abnormal data traffic within a short time, we should allocate as many packets as feasible to gateways until the accuracy and memory constraints are not satisfied. The remaining packets are transmitted to servers for inspection. By doing that, we can make use of benefits from gateways and servers.

We summarize the main contributions of our paper as follows.

- We first formulate the novel packet assignment optimization problem with the objective of minimizing detection time and constraints on detection and classification accuracy as well as the computational capacity.
- We analyze and compare the detection and classification performance of two different types of processing devices using state-of-the-art network traces.
- A novel and optimal packet assignment algorithm consisting of two phases (independent assignment and collaborative assignment phases) is proposed where each gateway makes a decision on which packets should be inspected. More specifically, the independent assignment phase allocates packets for gateways and servers based on the performance constraints and only the closest gateway to a sensor is responsible for attack detection of that sensor. Then, the collaborative assignment phase performs fine-tuning packet al.location by considering the computational capacity limitation of gateways. To achieve short detection time, packet assignment which exceeds the capacity of a gateway should be forwarded to nearby nodes.
- We design practical network scenarios and conduct the experiments with different parameters to verify the effectiveness of the proposed packet assignment algorithm.
- The experiments with different network parameters have been conducted and results show that our proposed method is able to obtain a close solution to the optimal one.

The rest of the paper is organized as follows. Section 2 introduces the network model and assumption considered in this study. Section 3 presents the problem formulation. Section 4 describes the proposed packet assignment algorithm. Section 5 analyzes the evaluation setting and results. Section 6 concludes the paper.

2 Network Model

We consider a network which consists of k sensors (or data sources), networking devices (e.g., gateways, switch), and external processing devices (e.g., server or cloud computers) as shown in Fig. 1. In this work, nodes are used to indicate networking devices and external processing devices. Two kinds of nodes are able to detect network attacks with different detection time and accuracy. Let us define a set of sensors $\mathbb{S} = \{1, 2, ..., k\}$ and a set of nodes $\mathbb{N} = \{1, 2, ..., n\}$. Sensor i generates and sends packets to end devices in the network for data storage with data rate r_i. Each gateway is connected to one or several external processing devices. Each node has specific features, e.g., the number of incoming packets per second, the maximal number of packets to be inspected per second, attack detection (task 1) capacity, attack classification (task 2) capacity denoted by $n_j^{in}, n_j^{max}, a_{j,1}, a_{j,2}$, respectively. We define v_j as the node type, e.g., $v_j = 1$ if node j is a gateway while $v_j = 2$ if node j is a server. Since there are two

types of nodes, we define $\mathbb{N}^1$ as a set of gateway nodes (type 1) and $\mathbb{N}^2$ as a set of server nodes (type 2).

A routing algorithm (e.g., hop count-based) is used to forward packets to destination addresses, which means routing paths for packets are determined in advance. We use $G(s_i)$ to indicate the closest gateway to sensor i which is in charge of forwarding packets of sensor i according to the routing algorithm. For example, $G(s_1) = \{2\}$ indicates that data collected by sensor i is first sent to gateway 2 before arriving at a destination device. In addition, t_{ij} denotes the detection time if node j is assigned to conduct the attack prediction task for data from sensor i ($t_{ij} > 0$ with $\forall j \in G(s_i)$). For gateway j, we define $S(n_j)$ and $G(n_j)$ as a set of sensors which have direct connection to node j and a set of servers which are affiliated with node j.

NIDS has two related tasks: attack detection (or anomaly detection) and attack classification. Nodes including gateways and servers are responsible for conducting the two tasks. However, we assume that gateway nodes can only perform the attack detection task due to the limitation of memory resources and computational capacity. Meanwhile, thanks to the large memory size and powerful processing units, server nodes can perform both detection and classification tasks. For better understanding of performance difference between gateways and servers, the evaluation section will compare the detection and classification accuracy using two different algorithms which are suitable for two types of processing nodes.

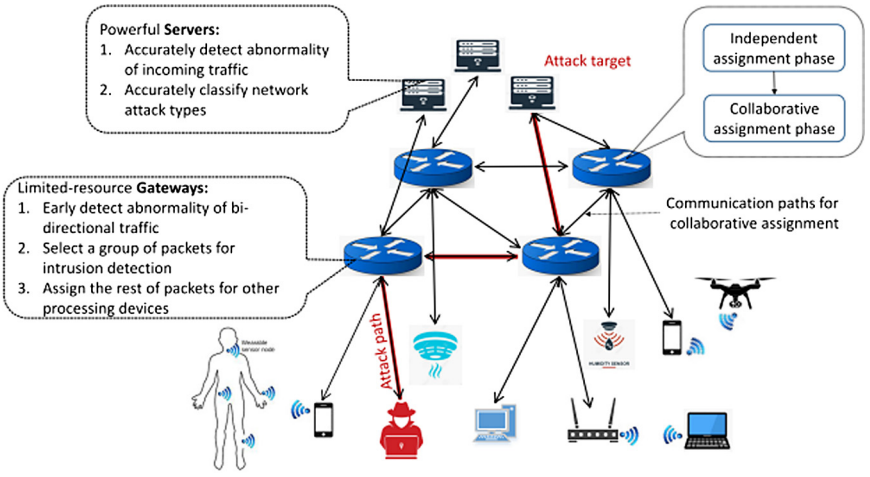

Fig. 1. Collaborative NIDS for interconnected networks with homogeneous devices

3 Problem Formulation

In this section, we present the formulation of an optimization problem for attack detection and classification with the given network model. The optimization problem is to minimize the detection time with performance and resource constraints of processing nodes. We first define decision variables as follows.

$$y_{ij} = \begin{cases} 1, & \text{if node } j \text{ is assigned to detect abnormal packets from } i^{th} \text{ sensor} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$z_{ij} = \begin{cases} 1, & \text{if node } j \text{ is assigned to classify an attack from } i^{th} \text{ sensor} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Table 1 summarizes definitions of notations used in the problem formulation.

Table 1. Model parameters

Notation	Description
k	The number of data generated devices
n	The number of processing nodes (gateways and servers) for NIDS
$\mathbb{S}$	A set of data generated devices
$\mathbb{N}$	A set of processing nodes (gateways and servers)
$\mathbb{N}^1$	A set of gateways nodes
$\mathbb{N}^2$	A set of servers nodes
r_i	Data rate of sensor i
n_j^{in}	The number of incoming packets per second at node j
n_j^{max}	The maximal number of packets to be inspected per second at node j
$a_{j,1}$	Attack detection capacity of node j
$a_{j,2}$	Attack classification capacity of node j
v_j	Node type
$G(s_i)$	The closest gateway to sensor i
$S(n_j)$	A set of sensors with direct connection to node j
$G(n_j)$	A set of gateways associated with node j
h_{ij}	The number of hop counts from sensor i to node j
t_{ij}	Detection time of node j for packets of sensor i

Based on the given routing paths in the network, we define h_{ij} as the number of hop counts from sensor i to node j. Assume that average hop-to-hop propagation time is t_0. The processing time at a hop is relatively small and

can be ignored. Therefore, detection time of sensor i's packets is estimated as $t(s_i) = t_0 \sum_{j=1}^{n} h_{ij} \times y_{ij}$ and the average detection time of the whole network is $\frac{t_0}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} h_{ij} \times y_{ij}$. The task assignment problem can be formulated as follows:

$$\min \quad \frac{t_0}{k} \sum_{i=1}^{k} \sum_{j=1}^{k} h_{ij} \times y_{ij} \tag{3}$$

subject to

$$\frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} y_{ij} a_{j,1} \geq a_d \tag{4}$$

$$\frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} a_{j,2} \geq a_c \tag{5}$$

$$\sum_{i=1}^{k} (y_{ij} + z_{ij}) r_i \leq n_j^{max}, \quad \forall j \in N \tag{6}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \in S \tag{7}$$

$$\sum_{j=1}^{n} z_{ij} = 1, \quad \forall i \in S \tag{8}$$

$$z_{ij} = 0, \quad \forall i \in S, j \in \mathbb{N}^1 \tag{9}$$

$$z_{ij} \geq y_{ij}, \quad \forall i \in S, j \in \mathbb{N}^2 \tag{10}$$

Equation (3) represents the objective function which aims to minimize the average detection time of all packets in the considered network. Inequalities (4) and (5) make sure that the average detection and classification accuracy should be at least the accuracy constraints a_d and a_c, respectively. Meanwhile, constraint (6) indicates that packet assignment should not violate the computational capacity of nodes. Equations (7) and (8) ensure that only one node is assigned for a specific task. Equation (9) guarantees that the intrusion classification task is not assigned to gateway nodes. At the same time, inequality (10) makes sure that a server node should perform both tasks if this node has already assigned to task 1. If $y_{ij} = 1$, z_{ij} should be set to 1. Inequality (10) is included in the optimization problem since we aim to minimize the packet transmission cost.

4 An Optimal Packet Assignment Algorithm

In this section, we will analyze the motivation of the proposed algorithm and describe the two-phase algorithm procedure in details as follows. First, based on the required detection accuracy, gateway j separately divides the incoming packets into two groups: group 1 processed by gateway nodes and group 2 inspected by server nodes. Let p_j denote the proportion of packets assigned for nodes in group 1 and then $(1 - p_j)$ is the proportion of packets for nodes in group 2. Assume that gateway and server nodes have detection performance a_1 and a_2, respectively ($a_{j,1} = a_1$ and $a_{j,2} = a_2, \forall j \in \mathbb{N}$). The average detection performance a_d can be estimated as $a_d = p_j a_1 + (1 - p_j)a_2$. Therefore, to achieve the required detection accuracy, we can determine packet grouping for gateway j as below.

$$p_j = \frac{a_2 - a_d}{a_2 - a_1} \tag{11}$$

Note that gateway j has limited processing capacity, n_j^{max}, which means $p_j n_j^{in} \leq n_j^{max}$. Then $p_j = \min(\frac{a_2 - a_d}{a_2 - a_1}, \frac{n_j^{max}}{n_j^{in}})$.

Based on the analysis of accuracy and bandwidth constraints, we design the optimal packet assignment algorithm which consists of two consecutive phases: independent assignment and collaborative assignment phases. As can be seen in Algorithm 1, in the first phase, gateway j assigns packets in a distributed manner. The proportion of packets inspected by nodes in group 1 p_j is estimated by using Eq. 11. Then, gateway j randomly selects traffic packets for intrusion detection with the probability p_j and allocates the detection task of the remaining packets to a server which is connected to gateway j. Since there can be multiple servers ($G(n_j)$) associated with gateway j, we assign a relatively similar load to these servers, i.e., $y_{ik} \sim B\left(1, \frac{1 - p_j}{|G(n_j)|}\right), \forall i \in S(n_j), k \in G(n_j)$. Note that notation $x \sim B(1, p)$ is used to indicate that random variable x follows the Bernoulli distribution with mean p. At the end of phase 1, detection tasks are assigned to gateways and servers based on the accuracy requirement. However, there may exist some gateways who could not perform all allocated tasks due to limited computing resource. Therefore, the proposed algorithm contains the second phase where the limited-resources gateways decide which tasks should be offloaded to nearby devices.

In the second phase, each gateway needs to check whether constraint on bandwidth is achieved or not. If $\sum_{i=1}^{k}(y_{ij} + z_{ij})r_i \leq n_j^{max}$, gateway j could not conduct intrusion detection on some packets, called uncompleted packets. Then, gateway j needs to find neighboring nodes which can perform the detection task on uncompleted packets. we use the hop count metric to measure the distance between two nodes. If there is a nearby gateway l who is willing to help gateway j and is closer to gateway j than the server connected to node j, we change task assignment of packet i from gateway j to gateway l, (i.e., $y_{ij} = 0$ and $y_{il} = 1$). Otherwise, the server k which is dedicated to gateway j is selected to perform the detection task on uncompleted packets.

Input: Sensor set $\mathbb{S}$, node set $\mathbb{N}$
Output: Assignment of two tasks to nodes: $y_{ij}, z_{ij}, \forall i \in \mathbb{S}, \forall j \in \mathbb{N}$
Initialize assignment:
$y_{ij} \leftarrow 0, z_{ij} \leftarrow 0, \forall i \in \mathbb{S}, \forall j \in \mathbb{N}$
Independent assignment phase:
for $j \in \mathbb{N}^1$ **do**
  $p_j \leftarrow \dfrac{a_2 - a_d}{a_2 - a_1}$ /* compute p_j using constraints on accuracy */
  $y_{ij} \sim B(1, p_j), \quad \forall i \in S(n_j)$ /* assignment to gateway j follows Bernoulli distribution with mean p_j */
  $y_{ik} \sim B\left(1, \dfrac{1-p_j}{|G(n_j)|}\right), \quad \forall i \in S(n_j), \forall k \in G(n_j)$ /* assignment to server k has Bernoulli distribution with probability $\dfrac{1-p_j}{|G(n_j)|}$ */
end
Collaborative assignment phase:
for $j \in \mathbb{N}^1$ **do**
  $i \leftarrow 1$
  if $(y_{ij} = 1)$ & $(i \leq k)$ & $\left(\sum_{i=1}^{k}(y_{ij} + z_{ij})r_i \geq n_j^{max}\right)$ **then**
    $y_{ij} \leftarrow 0$ /* remove task assignment for gateway j */
    for $l \in \mathbb{N}^1$ $(l \neq j)$ **do**
      if $h_{il} \leq h_{ik}$ *with* $k \in G(n_j)$ **then**
        $y_{il} \leftarrow 1$ /* l is a close gateway to node j */
      end
      else
        $y_{ik} \leftarrow 1$ /* k is a server associated with node j */
      end
    end
  end
  $i \leftarrow i + 1$
end

Algorithm 1: An Optimal Packet Assignment Algorithm

5 Performance Evaluation

5.1 Experimental Setup

In this subsection, we describe a practical network scenario and network traces for evaluation of the proposed algorithm. We consider a heterogeneous network including sensors from different applications (e.g., smart home, smart healthcare, smart city, smart agriculture). In each application, sensors are deployed and connected in a sub-network, e.g.., 28 different sensors scattered around a smart home [2,7]. Assume that each gateway is responsible to forward data of several

sub-networks. The parameters of network scenarios are summarized in Table 1. For edge devices, we consider Virtex-7 FPGA-based gateways since the Virtex-7 FPGA board [13] with 64 Mb RAM is the most advanced experimental kit manufactured by Xilinx. Each gateway can have 4, 8 Ethernet ports where a half of them is used for incoming traffic and the remaining ports are dedicated for outgoing data. Assume that each port of gateways can support data rate at 100 Mbps.

In this work, the UNSW-NB15 dataset is selected to evaluate the performance of the proposed assignment algorithm. The UNSW-NB15 dataset includes real normal and synthetic abnormal network traffic traces during a 16-hour experimental period and consists of 9 attack classes. For our evaluation, the full dataset is divided into a training data set of 175,341 samples and a test data set of 82,332 samples, and one third of the training data set is used as the validation dataset. The validation set is used to find the best hyper-parameters for the prediction model. In this traffic trace, around 68% of samples are normal while the remaining data belongs to packets associated with one of 9 network attacks. The UNSW-NB15 dataset provides 48 traffic features for each data connection. We convert these features into 196 numeric attributes for the intrusion detection model.

Table 2. Network scenarios

Parameters	Values
The number of gateways	5
The number of servers	2
The number of sub-networks	{15, 21}
Processing capacity of gateways	0.5
The number of available ports at each gateway	{4, 6 }
Data rate of each sub-network	100 Mbps
Intrusion detection accuracy requirement	73%

5.2 Performance Comparison of Processing Nodes

We select and compare performance of detection algorithms which are suitable for gateways and servers based on their memory size and computing capacity. Due to the resource limitation, a logistic regression-based intrusion detection method is constructed for gateway nodes. Meanwhile, in order to gain improvement over the logistic regression-based model, we use an autoencoder model, which learns the latent representation of traffic packet, followed by a fully connected neural network-based intrusion detection model for cloud nodes. The performance of two constructed models are summarized in Table 3. We compare two models in terms of the number of training parameters and detection accuracy. The logistic regression-based detection model which consists of a significantly smaller number of training parameters and produces less accuracy (with

roughly 16% difference) compared the neural network-based attack detection model. For the evaluation of the proposed packet assignment algorithm in the next subsection, we use the collected accuracy of the logistic regression-based and neural network-based models as the detection accuracy of gateways and servers.

Table 3. Performance of prediction models

Models	No. of input features	No. of parameters	Accuracy (%)
LR-based detection	196	197	70.62
NN-based detection	196	25,791	86.3
NN-based classification	196	25,980	73.37

5.3 Evaluation of the Proposed Packet Assignment Algorithm

In this subsection, we describe a small example to show how the proposed packet assignment algorithm updates the solution at different phases. Then, using a larger network scenario as shown in Table 2, the proposed algorithm is verified and compared to the optimal solution in terms of detection accuracy and time.

Figure 2 demonstrates the packet assignment when the number of sub-networks and nodes are set to 10 and 5, respectively. Among 5 processing nodes, the first three nodes are gateways and the last two nodes are servers. Gateways 1 and 2 have connected to four sub-networks while gateway 3 is responsible for two sub-networks. The processing capacity of gateways is set to 0.5, i.e., each gateway should only perform network intrusion detection on 50% of sub-networks. Assume that each gateway consists of 4 networking ports then only 2 sub-network flows can be inspected in each gateway. We use a binary matrix consisting of ten rows and five columns to represent the assignment solution. The value at the i^{th} row and j^{th} column represents the assignment of node j to sub-network i, e.g., value 1 means assignment and 0 indicates no assignment. At the initial phase, all values of assignment matrix are set to 0, i.e., there is no assignment for processing nodes.

Then, in the next phase, the proportion of packets which are inspected by each gateway is estimated. In this example, $p_j = \dfrac{a_2 - a_d}{a_2 - a_1} = \dfrac{86.3 - 73}{86.3 - 70.62} = 0.84$. Then, the number of sub-networks which are inspected by gateways 1 and 2 is $\lfloor 0.84 * 4 \rfloor = 3$. Similarly, gateway 3 should perform anomaly detection on one sub-network in order to satisfy the detection accuracy requirement. Since there are four sub-networks which forward data packets to gateway 1 and gateway 1 should detect intrusion for 3 sub-networks due to the accuracy constraint, gateway 1 simply selects the first three sub-networks for intrusion detection and sends data from the fourth sub-network to node 4 (server 1). After the independent assignment phase has completed, anomaly detection for packets of ten sub-networks are assigned to five nodes considering the accuracy requirement as shown in Fig. 2(b).

Finally, the collaborative assignment phase implements fine-tuning the current assignment based on the computing limitation of gateways in order to minimize the detection time. Assume that each gateway can perform the intrusion detection function for two sub-networks due to the limited memory and computing capacity. As a result, gateway 1 needs to ask the neighboring node to conduct the detection task for one sub-network. As can be seen in Fig. 2(c), gateway 1 selects and forwards traffic data of the third sub-network to node 4 (server 1). Similarly, gateway 2 transmits packets of the seventh sub-network to node 5 (server 2) for intrusion detection.

In order to evaluate the effectiveness of our proposed algorithm, we made performance comparison between the proposed algorithm and the optimal solution. Note that the optimal solution is obtained by using the brute-force search while the proposed algorithm determines a packet assignment solution based on a distributed manner. As can be seen in Table 4, the proposed assignment method produces similar detection time to the optimal solution in both cases of 15 and

Table 4. Performance comparison with the optimal solution

	Proposed algorithm	Optimal brute-force-based algorithm
Average detection time (15 sub-networks)	$6.93t_0$	$6.86t_0$
Detection accuracy (15 sub-networks)(15 sub-networks)	75.84%	75.84%
Average detection time (21 sub-networks)	$6.67t_0$	$6.67t_0$
Detection accuracy (21 sub-networks)	75.1%	75.1%

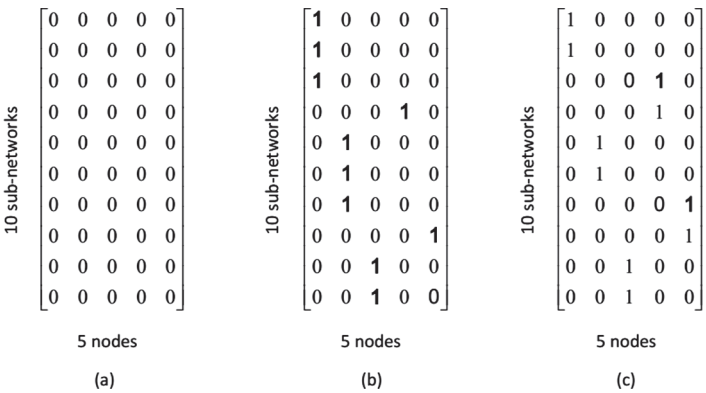

Fig. 2. Packet assignment at different phases of the proposed algorithm, (a): Initial assignment, (b): Independent assignment, and (c): Collaborative assignment

21 sub-networks. For instance, if there are 15 sub-networks, average detection time of our method and the optimal solution is $6.93t_0$ and $6.86t_0$, respectively (t_0 is the one-hop propagation time).

6 Conclusion

In this paper, we address the problem of balancing between detection time and detection accuracy in a multi-level network intrusion detection system. In this system, both fog and cloud devices are assigned tasks of anomaly detection and classification in order to shorten the detection time while satisfying the accuracy requirement. We define the novel optimization problem which aims to minimize the detection time with constraints on accuracy and computational capacity of computing nodes. Then, a heuristic packet assignment method is proposed to find a solution for the optimization problem. The proposed method is evaluated by using the UNSW-NB15 dataset and the results show promising performance in terms of detection time and accuracy. As a future work, we plan to design an intelligent and self-adaptive packet assignment algorithm which is able to learn optimal solutions given the change of network conditions (e.g., the number of sub-networks and the number of neighboring gateways).

Acknowledgment. This work is funded by the Le Quy Don Technical University.

References

1. Cao, V.L., Nicolau, M., McDermott, J.: Learning neural representations for network anomaly detection. IEEE Trans. Cybern. **49**(8), 3074–3087 (2019)
2. Clark, M., Dutta, P.: The haunted house: Networking smart homes to enable casual long-distance social interactions. In: IoT-App 2015 (2015)
3. Doffman, Z.: Cyberattacks on IOT devices surge 300% in 2019, 'measured in billions', report claims (2019). https://bit.ly/35uPCI7. Accessed 04 May 2020
4. Hosseini, S., Azizi, M.: The hybrid technique for DDoS detection with supervised learning algorithms. Comput. Netw. **158**, 35–45 (2019)
5. Khan, F.A., Gumaei, A., Derhab, A., Hussain, A.: A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access **7**, 30373–30385 (2019)
6. Lapolli, A.C., Marques, J.A., Gaspary, L.P.: Offloading real-time DDoS attack detection to programmable data planes. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 19–27 (2019)
7. Morais, C., Sadok, D., Kelner, J.: An IoT sensor and scenario survey for data researchers. J. Braz. Comput. Soc. **25**, 4 (2019)
8. Moustafa, N., Turnbull, B., Choo, K.R.: An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet Things J. **6**(3), 4815–4830 (2019)
9. Carvalho, R.N., Bordim, J.L., Alchieri, E.A.P: Entropy-based dos attack identification in SDN. In: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 627–634 (2019)

10. Nguyen, T.G., Phan, T.V., Nguyen, B.T., So-In, C., Baig, Z.A., Sanguanpong, S.: Search: a collaborative and intelligent NIDS architecture for SDN-based cloud IoT networks. IEEE Access **7**, 107678–107694 (2019)
11. Systems, C.: Cisco Annual Internet Report (2018–2023) White Paper. Technical Report Cisco Systems (2020)
12. Vu, L., Cao, V.L., Uy, N.Q., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E.: Learning latent distribution for distinguishing network traffic in intrusion detection system, pp. 1–6 (2019)
13. Xilinx: Xilinx Virtex-7 FPGA VC707 Evaluation Kit. Tech. rep., Xilinx
14. Yan, Q., Huang, W., Luo, X., Gong, Q., Yu, F.R.: A multi-level DDos mitigation framework for the industrial internet of things. IEEE Commun. Mag. **56**(2), 30–36 (2018)
15. Yang, Y., Zheng, K., Wu, C., Yang, Y.: Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors **19**(11), 2528 (2019)