BIDS: Bio-Inspired, Collaborative Intrusion Detection for Software Defined Networks

Qianru Zhou
School of Computing Science
University of Glasgow
Glasgow, UK
qianru.zhou@glasgow.ac.uk

Dimitrios P. Pezaros
School of Computing Science
University of Glasgow
Glasgow, UK
dimitrios.pezaros@glasgow.ac.uk

***Abstract*—With network attacks becoming more sophisticated and unpredictable, detecting their onset and mitigating their effects in an automated manner become increasingly challenging. Lightweight and agile detection mechanisms that are able to detect zero-day attacks are in great need. High true-negative rate and low false-positive rate are the most important indicators for a intrusion detection system. In this paper, we exploit the logically-centralised view of Software-Defined Networking (SDN) to increase true-negative rate and lower false-positive rate in a intrusion detection system based on the Artificial Immune System (AIS). We propose the use of an *antibody fuser* in the controller to merge and fuse the mature antibody sets trained in the individual switches and turn the real intrusion records each switch has seen into antibodies. Our results show that both the false-positive rate and true-negative rate experience significant improvement with the number of local antibody sets fused grows, consuming less cpu usage overhead. A peak improvement can reach over 80% when antibody sets from all switches are taken into consideration.**

***Index Terms*—Artificial immune system, Intrusion detection, Software Defined Networking**

I. INTRODUCTION

Cyber security is gaining significant traction as we increasingly rely on ICT infrastructures for industry, economics, communication, and other domains. One of the most important fields in cyber security is anomaly and intrusion detection, which deals with defining normal network behaviour, and subsequently identifying deviations from that pattern.

The Artificial Immune System (AIS) which is inspired by the observed mammalian immune process is naturally tuned to the problem of detecting anomalies. By detecting certain activities that deviate from what has been defined as normal, the AIS exhibits many virtues that are highly desired by a network anomaly detection system. These include: autonomy, robustness, light-weightiness, and adaptivity [1]. Hence, artificial immune systems have been explored for network intrusion detection [1], [2], [3], [4].

By abstracting network management operations into an centralised control plane, Software Defined Networking (SDN) is able to construct a global knowledge base that enables more advantaged security management. Various approaches have been proposed using SDN for intrusion detection, bringing the following benefits: 1) With the programmable control plane, it is easier to collect flow-related traffic features [5]; 2) based on SDN's centralised control, it is more convenient to collect information and do network statistics analysis [6]; 3) With the global view of network traffic, we can orchestrate the decision making strategy, and achieve higher accuracy while avoiding false positives [7]. It is worth mentioning that, although currently every network adopting OpenFlow is deemed as a deployment of SDN, SDN itself has wider scope than that. OpenFlow is just one prototype protocol realisation of SDN [8].

In this paper, we propose BIDS (Bio-Inspired, collaborative intrusion Detection for SDN), an AIS-based network Intrusion Detection System (IDS) to exploit the centralised orchestration of SDN. The distributed AIS IDS is running on each switch in a network, while an *antibody fuser* running on the controller orchestrates the training results from all the switches. At the beginning, each IDS component is trained by the local traffic seen at the switch. After the training process, the mature antibody sets trained and the intrusion set recorded by each switch will be collected by the controller, where they will be fused by the *antibody fuser* as one global mature antibody set and subsequently synced with all switches. Thus, with knowledge of global traffic and intrusions experienced throughout the network, the false positive rate will be reduced while the true-negative rate increases.

The reminder of this paper is structured as follows: Section II provides the state of art in AIS-based intrusion detection. We detail the architecture of the proposed system in Section III. Experimental design and evaluation results are presented in Section IV. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

An AIS is an abstraction of the sophisticated mammalian immunological process which is still developing to date. The mapping between the mammalian immune process and network intrusion detection based on AIS can be seen in Table I. The network is compared to the human body while switches are the lymph glands. The gene segment patterns of antibodies and antigens are modelled as binary strings with fixed length l. Thus, the *self and non-self discrimination* is defined by the *matching rule* of binary strings. Matching rules are key for AIS, since they determine the matching process.

Since 1994, when the AIS was first explored for network security, the fundamental algorithm has not changed much [1], [2], [3], [4], [9]. However, various frameworks have been proposed by different groups. An ARTificial Immune System (ARTIS) was proposed by Hofmeyr et al. [1] in 1999. It abstracts TCP traffic into a "datapath triple" – "source IP

TABLE I
THE MAPPING BETWEEN MAMMALIAN IMMUNE SYSTEM TO ARTIFICIAL IMMUNE NETWORK

Mammalian Immune System	Network Intrusion Detection System
Human Body	Network
Lymph Gland	Switch
Antigen	Binary strings extracted from IP packets
Cell B, T, and Antibody	Binary strings
Binding between Antibody and Antigen	R-contiguous rule
Toleration	Negative selection

address, destination IP address, TCP service type", represented by a 49-bit binary string. Based on previous work, Hosseinpour et al. [3] proposed an evolved framework, distributing the training and detection function to each switch as an agent of the central intrusion detection engine which is located on the server and only process the initial detection result reported by switches to make final detection decisions. The antibodies and antigens they designed are 112-bits binary strings, containing the following information from TCP packets: source and destination IP address, source and destination port number, packet length, and protocol.

To our knowledge, there has not been any work on AIS-based IDS for SDN. However, intrusion detection for SDN has been widely investigated, and most of the proposed approaches are based on the flow tuple. In [11], the authors proposed BroFlow, an policy-based IDS that takes advantage of OpenFlow APIs. By collecting flow statistics and calculating the entropies, Giotis et al. proposed an IDS to detect massive DDoS over SDN [5]. Tang et al. proposed a deep learning model based IDS for SDN [13]. Even though our work is based on previous work by Forrest et al. [4] and Hosseinpour et al. [3], there are several implementations that are different, as shown below.

- In the antibody training process, not only is normal traffic data collected as a self set, but the history of intrusions experienced by the switch are also collected as the local intrusion set. The immature antibodies will be trained by both the self and the intrusion sets to reduce the false-positive rate.
- The local mature antibodies generated after the training process in each switch are collected by the controller, where they are fused and merged to a final antibody set. This antibody set will be synced with all the switches and work as the detector for future intrusions. Once a match is found, an alert of intrusion will be send to the network admin.

III. AIS-BASED INTRUSION DETECTION

In this section, we describe our proposed BIDS based on AIS. The main purpose of the proposed IDS is to take advantage of the global control of SDN to reduce false-positive rates without affecting true-negative rates. Fig. 1 shows the fundamental design of BIDS. The *antibody generator* and the *trainer* are located at each switch, while the antibody sets generated are collected and fused by the controller and then synced back with the switches again.

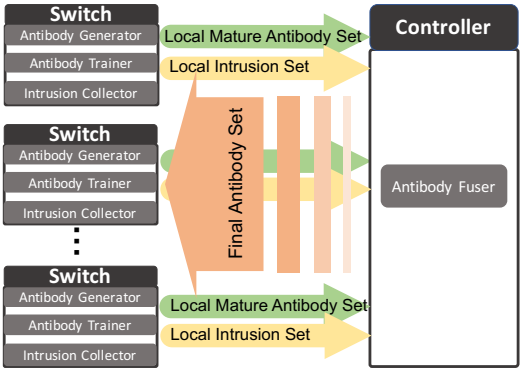

Fig. 1. The proposed AIS based intrusion detection system BIDS for SDN.

The procedure of BIDS is shown in Fig. 2. The detail of each component is discussed below.

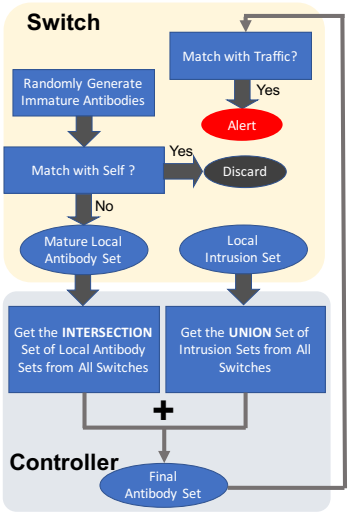

Fig. 2. Detection procedure flowchart of BIDS.

A. Antibody Generator

In the antibody generator, binary strings with fixed length l are generated randomly as the immature antibodies. To extract the patterns of network traffic, information from TCP packets are converted to binary strings. After studying the pattern of real intrusion packets from the NSL-KDD dataset[1], we choose the following information: source IP address, source port number, destination IP address, destination port number, application protocol, type of service, and packet length, as shown in Table II, constructing a $l = 136$ bits binary string. For the intrusion types recorded in the NSL-KDD dataset, these fields experience the most significant deviation between intrusion packets and normal packets. Theoretically, all the information in a packet can be used as the signature. However, as the storage and computational cost grow exponentially with the length of the binary string l, it is not feasible to add all the information that can potentially denote the abnormality.

[1]The following types of intrusion data in NSL-KDD dataset are used: *back, buffer_overflow, guess_passwd, ipsweep, loadmodule, neptune, nmap, pod, portsweep, rootkit, satan, smurf, teardrop, warezclient*.

TABLE II
INFORMATION EXTRACTED FROM IP PACKETS

Name of Field	Binary String Length
Source IP Address	32 bits
Source Port	16 bits
Destination IP Address	32 bits
Destination Port	16 bits
Application Protocol	16 bits
Type of Service	8 bits
Packet Length	16 bits
Total	136 bits

B. Antibody Trainer

Negative selection is performed by the *antibody trainer*. The immature antibodies will be compared with normal traffic datasets collected locally by the switch, which is also a set of binary 136-bit-long strings. Similar to previous work [3], [1], assume that during the traffic collection period no intrusion happens (as in our production network environment, an intrusion is a low probability event). Whenever an immature antibody is matched with normal traffic, it is extinguished and only those who do not match any normal traffic strings remain as the mature antibodies.

C. Intrusion Collector

To achieve more accurate detection, we collect real intrusion data from the NSL-KDD dataset, and extract the same information as the antibodies into binary strings as the intrusion set. We directly append the non-self set to the antibody set in the controller through the *antibody fuser* discussed below. As the detection mechanism in AIS-based intrusion is as simple as binary string matching, putting real intrusion data in as antibodies will make the string matching more efficient.

D. Antibody Fuser

Once the collection of local mature antibody set A_i and intrusion set I_i in each switch i are complete, the SDN controller will send them to the *antibody fuser*, which is responsible for carrying out the following three tasks.

- Take the intersection of the mature antibody sets from all switches $\bigcap_i A_i$.
- Take the union of the intrusion sets from all switches $\bigcup_i N_i$.
- Unite the two sets from the previous two steps, to create the *final antibody* set FA and dispatch to all the switches.

$$FA = (\bigcap_i A_i) \cup (\bigcup_i N_i)$$

We argue that the antibody fuse process can reduce the false-positive rate, and increase the true-negative rate. As each user has personalised online behaviour, the traffic patterns vary between different switches. As shown in Fig. 3, the rectangle in each subfigure is a two-dimensional representation of the universal set of 136 bits binary strings U. The strings are classified into two categories, either *self* (represented by the white area) S or *non-self* (represented by the grey area) N, i.e., $U = S \cup N, S \cap N = \oslash$. Let the traffic experienced by each switch i be T_i, in which self set is S_i and non-self set is N_i. We have

$$T_i = S_i + N_i,$$

$$U = \sum_{i=1}^{n} T_i,$$

where n is the total number of switches in the network.

To simplify the calculation, we assume that the traffic experienced by different switches has no overlap, which means $T_i \cap T_j = \oslash, S_i \cap S_j = \oslash, N_i \cap N_j = \oslash, i \neq j$.

The antibody set trained by switch i is $A_i = U - S_i$

The false positive rate generated using A_i is

$$FP_i = \frac{S - S_i}{U}$$

After fusing the antibody sets trained by switch $1, 2, ...n$, the antibody set generated will be

$$A_{1...n} = U - \sum_{i=1}^{n} S_i$$

The false positive rate $FP_{1...n}$ using the fused antibody set $A_{1...n}$ is

$$FP_{1...n} = \frac{A_{1...n} - I}{U} = \frac{S - \sum_{i=1}^{n} S_i}{U} = \frac{S}{U} - \frac{\sum_{i=1}^{n} S_i}{U}$$

Thus, the false positive rate after using *antibody fuser* is

$$FP_{1...n} = \gamma - \frac{\sum_{i=1}^{n} S_i}{U}$$

where $\gamma = \frac{S}{U}$ is a constant. Obviously, $FP_{1...n}$ decreases with the number of fused antibody sets n.

A two-dimentional representation of the fusing process is shown in Fig. 3(a), the local traffic experienced by switch 1 T_1 is represented by the yellow ellipse, most of which is normal traffic (the overlap region with the white area), while the rest are known intrusions N_1 (the area overlapped with grey area). In the local IDS in switch 1, the training set is represented as area labelled "1" in Fig. 3(b). As a result, the antibody set A_1 trained by switch 1 is the shadowed area. The area overlapped with the white zone, which is labelled in "2", is the false-positive results. In other words, the traffic strings in zone "2" is normal traffic, but switch 1 will deem them as abnormal because it has not seen these strings before, and hence false alarms will be issued for them. As shown in Fig. 3(b), the false-positive rate is relatively high for switch 1, due to its limited training set, or in other words, because it has not seen much of the superset of traffic.

The traffic of switch 2 is shown in Fig. 3(c) which covers a different area from switch 1. With the antibody fuser, the antibody and intrusion sets of switch 1 & 2 can be fused $(A_1 \cap A_2) \cap (I_1 \cup I_2)$, as shown in the shadow area of Fig. 3(d). The false-positive rate has shrunk to the area labelled with "4". From here we can see that, with more different antibody sets fused, the false-positive rate will be reduced.

IV. EVALUATION

A. Data Preparation

1) Immature Antibody Set: A number of 136-bit binary strings were generated randomly as the *immature antibody*. In theory, the *immature antibody set* should be an exhaustive

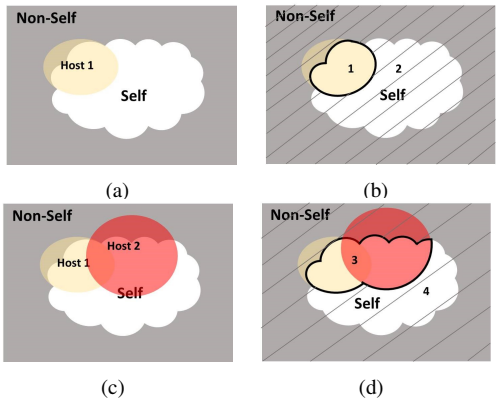

Fig. 3. The fuse process of two antibody sets in two-dimensional representation of the universe of 136 bits binary strings.

list of all possible 136-bit binary strings. However, that will be a total of 2^{136} binary strings, which is impossible to be stored or computed in reasonable time with today's technology. Thus, we generate a relatively large number (compared to the training dataset) – 5 million unique binary strings, as the immature antibody set for the experiment.

2) Self Data Set: The self training dataset used for this work is collected from real traffic. The self sets were collected by four switches with different online behaviour preference for 5 days, during which a total of 65,328 unique TCP packets were logged and converted to 136 bits binary strings. The size of each self set is shown in Table III.

TABLE III
FEATURES OF TRAINING SETS.

switch	Number Intrusion Strings	Number Self Strings
switch 1	2423	11790
switch 2	2423	13239
switch 3	2423	12979
switch 4	2423	27320
Total	9702	65328

3) Intrusion Data Set: As discussed broadly, the DARPA[2] and KDD'99[3] datasets are not suitable to serve as viable training datasets [10]. Thus, an advanced dataset NSL-KDD[4] is used as the intrusion training dataset. A total of 9702 intrusion strings were extracted from the dataset and were dispatched evenly across four switches. Thus, there are 2423 strings in the local intrusion set from each switch. Table III gives a breakdown of the number of training strings used in each switch.

To test the true-negative rate of BIDS , real *port scanning* attacks launched by Nmap[5] are recorded and mapped into 136-bit binary strings. A total of 8895 binary strings are collected as the intrusion test set.

B. Matching Rules Setup

There are many string matching rules, some obvious ones are Hamming distance, edit distance, and r-contiguous bits. In this paper, the more immunologically plausible *r-contiguous bits rule* is adopted. In the r-contiguous bits rule, two strings with the same length match if they have at least r contiguous identical bits. The threshold r determines the minimum affinity of two strings given. If the threshold r is too small, the rule will fail to distinguish the intrusion string from intrusion string, and thus result in a high false-positive rate. When r is too big, the matching time will grow exponentially, besides, it will be more difficult to match the intrusion string with the antibody string, which results a high false-negative rate. In this work, different values of r are examined during the evaluation experiment, performing the following matches:

- Match the test set of intrusion (non-self) strings with the test set of self strings;
- Match two test sets of self strings collected at different time slots by the same switch.

According to our experiment, when the threshold r is any value lower than 26, the number of matched strings in the self and non-self set is overwhelming. In an extrema case, when $r \leq 18$, the self and non-self sets match 100%. When $r \geq 26$, most of the self strings fail to match with non-self strings. Thus, the value $r = 26$ is chosen for the r-contiguous matching rule in the experiment. There are reports investigating more appropriate rules for the AIS-based intrusion detection, but it is beyond the scope of this paper.

C. Results

The results of the experiments are presented in this section. The performance of BIDS is mainly evaluated in terms of two metrics: *false-positive* and *true-negative*. *False-positive* occurs when a normal string is classified as intrusion, while *true-negative* is the situation that an alarm is generated as soon as a intrusion takes place [1].

The results calculated by BIDS with and without the use of the *antibody fuser* are compared. Comparison with previous AIS-based IDS is presented in detail. The computational cost of BIDS using the *antibody fuser* is also compared against other popular IDS proposed for SDN. The main purpose of the experiments is to demonstrate that the use of centralised SDN control – *antibody fuser* improves the accuracy of BIDS, both in terms of *false-positive* and *true-negative*.

TABLE IV
FALSE POSITIVE RATE OF ALL TEST SETS USING FUSED ANTIBODY SETS WITH ALL POSSIBLE COMBINATIONS.

Fused Antibody Sets	Self Set 1	Self Set 2	Self Set 3	Self Set 4
$A_1 \cap A_2$	0.00%	0.00%	88.64%	87.57%
$A_2 \cap A_3$	86.64%	0.00%	0.00%	86.49%
$A_3 \cap A_4$	90.00%	89.78%	0.00%	0.00%
$A_1 \cap A_3$	0.00%	91.03%	0.00%	92.12%
$A_2 \cap A_4$	86.28%	0.00%	70.03%	0.00%
$A_1 \cap A_4$	0.00%	86.91%	84.55%	0.00%
$A_1 \cap A_2 \cap A_3$	0.00%	0.00%	0.00%	85.96%
$A_2 \cap A_3 \cap A_4$	82.45%	0.00%	0.00%	0.00%
$A_1 \cap A_2 \cap A_4$	0.00%	0.00%	58.06%	0.00%
$A_1 \cap A_3 \cap A_4$	0.00%	78.70%	0.00%	0.00%
$A_1 \cap A_2 \cap A_3 \cap A_4$	0.00%	0.00%	0.00%	0.00%

1) False-Positive Rate: We have fused the mature antibody sets trained by the 4 switches (noted as A_1, A_2, A_3, and A_4) with all possible combinations. As the self sets collected

[2]https://www.ll.mit.edu/r-d/datasets
[3]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[4]http://www.unb.ca/cic/datasets/nsl.html
It is believed to have fixed some problems of KDD'99.
[5]http://nmap.org/

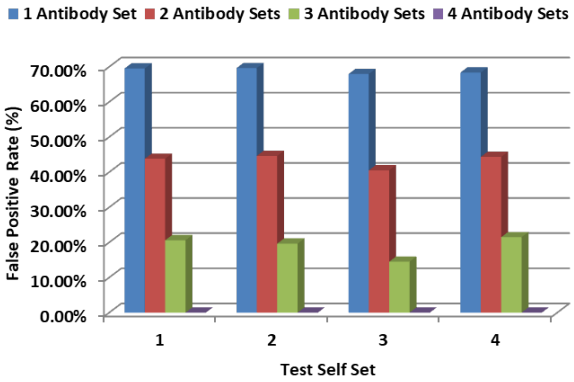

Fig. 4. False-positive rate calculated by using various combinations of fused antibody sets test on four self set collected by four switches.

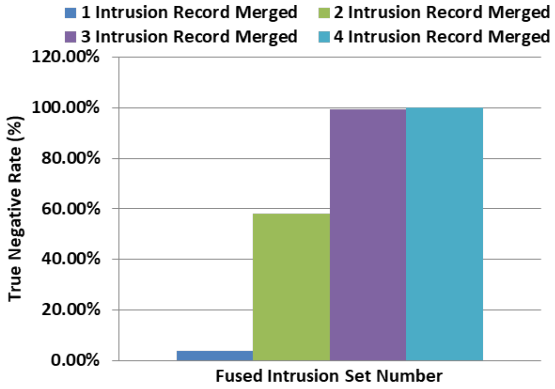

Fig. 5. True-negative rate calculated by using different number of fused antibody sets test on intrusion set collected.

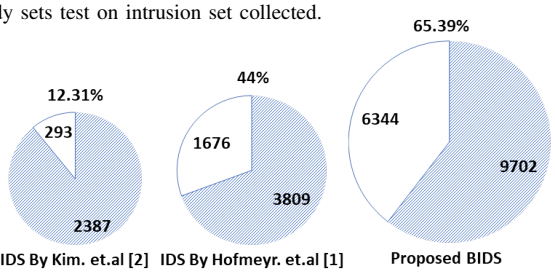

Fig. 6. True-negative rate comparison with BIDS and related work.

in different switches (which work as the training datasets) experience different patterns, the IDS is expected to produce high false-positive rate when the local antibody set trained in one switch is tested by the self set collected by another switch. The experimental results are listed in Table IV. The four self sets collected by the four switches have been tested on all possible combinations of fused mature antibody sets. In the column "Fused Antibody Set", the antibody sets fused are presented. The average false-positive rates of the antibody set fusing different number of local antibody sets are presented as bar charts in Fig. 4. The X-axis of Fig. 4 represents the four test datasets collected from the four switches, and the legend represent the numbers of antibody sets that are fused. It is worth mentioning that the blue bars on the far left of the results, represent the results tested by a single local antibody set, without using the *antibody fuser*. The Y-axis in Fig. 4 is the false-positive rate calculated as a percentage.

In contrast to the results testing on the single antibody set (the blue bar on the far left of each self set experiment), the fused antibody sets can significantly reduce the false-positive rate. As obviously shown by the trend of result as the number of fused sets grow, the false-positive rate reduces significantly with more and more local antibody sets are fused. In an extreme case, when all the switches' antibody sets are fused, as shown by the purple bar on the far right end in all results, the false-positive rate drops to 0. This is because with all the local antibody sets being fused, the fused antibody set has been trained by all the self sets, and will not make false positive decisions. We argue that although the training sets of the switches are used to test the fused antibody set, which generates a little bit ideal result such as 0% false-positive rate when all the local antibody sets are fused, the trend shown in Fig. 4 demonstrates the decline on the false-positive rate with our BIDS algorithm. It evident that in an artificial immune system based intrusion detection, which entirely depends on learning what has been defined as normal and detect anything that deviates from normality, the global view provided by SDN can reduce the false-positive rate caused by the limitation on the vision of local traffic pattern.

2) True-Negative Rate: The four intrusion records are merged to the antibody set fused in the above section. The final antibody set is tested over the intrusion test set collected from real *port scanning* attacks performed by Nmap. The results are presented in Fig. 5. The legend shows the number of intrusion sets fused. The Y-axis is the percentage of intrusions successfully detected. The bar at the far left is the true-negative rate before any intrusion sets have been fused. Apparently, the true-negative rate is very low, due to the limited immature antibodies used in the training process (as stated in Section IV-A1, we could not afford using a exhaustive list of all possible 136-bit binary strings, thus we cannot achieve a high true-negative rate of real intrusions before any intrusion training set is fused). However, the intrusion true-negative rate soars with the number of intrusion sets fused to the antibody set increases. When more than half of the switches' intrusion sets are fused, BIDS is able to detected the majority of the intrusions strings tested. The improvement on true-negative rate by using *antibody fuser* is evident.

Comparison between BIDS and previous work on IDS using artificial immune system [1], [2] is presented with pie chart in Fig. 6. True-negative rates in previous work are the average value obtained from all tests reported. The true-negative rate of BIDS is obtained by taking the average of the tests result using 2, 3, and 4 fused intrusion records. The number in the pie's shadow area denotes the number of binary strings in the intrusion set used, while the number in the blank area is the the number of strings detected. As it can be seen from Fig. 6, not only a much larger intrusion set is used, BIDS also outperforms previous work with the fusion of intrusion sets. As shown in the far left pie in Fig. 6, the true-negative rate of IDS by Kim et al. is quite low with only 12.31%, due to the inappropriate threshold and the fact that the antibody set is too small [2]. Results from the IDS proposed by Hofmeyr et al., as shown in the middle pie in Fig. 6, demonstrate a much

better performance, reaching 44% [1]. This is probably due to the fact that, as claimed by the authors, an exhaustive list of all possible binary strings was generated as the immature antibody set for the training process. As for our results, as shown in the far right pie in Fig. 6, although we could not afford to use an exhaustive list of all possible binary strings as the immature antibody set, BIDS achieves the highest average true-negative rate.

TABLE V
COMPARISON OF CPU USAGE FOR PREVIOUS IDS FOR SDN AND BIDS.

IDSs for SDN	CPU Usage (%)
Entropy based with sFlow [5]	39%
TRW-CB with sFlow [12]	39%
Entropy based with Native OF [5]	61%
TRW-CB with Native OF [12]	58%
BroFlow when 5000 packets/s [11]	100%
Proposed IDS – BIDS	**15.3%**

3) CPU Usage: Computational cost is a key performance indicator for any IDS, especially when this runs in-network. In Table V, we compare the average CPU usage of BIDS against related IDS for SDN that we discussed earlier in the paper [5], [12], [11]. These proposed SDN IDS are based on flow operations, and are target on one specific intrusion type only, DoS. As AIS based IDS could detect any deviance from normal traffic, thus, it is able to detect a wide spectrum of intrusions, including "Zero-day" attacks. Besides, as the AIS algorithm works by performing binary string matching, it is more efficient than the other machine learning algorithms such as deep learning, as can be evident by the result of [5] and BIDS shown in Table V. Generally, BIDS performs better than the other SDN IDS. The light-weightness of AIS algorithms is proven.

V. CONCLUSIONS

In this paper, we have presented an Artificial Immune System (AIS)-based IDS for Software Defined Networks (SDN) – BIDS . Based on traditional AIS intrusion detection, it takes advantage of the global network view provided by SDN to collect and fuse the mature antibody sets trained and the intrusions recoded by the local switches, to achieve lower false-positive rates and higher true-negative rate.

The contributions of BIDS have been demonstrated over a prototype testbed implementation. The analysed normal dataset was gathered from real traffic in a research office environment. During 5 days, network traffic was generated following different online behaviour and collected from 4 switches, free from any known intrusions. The intrusion training set was generated from the NDL-KDD dataset, which contains 14 kinds of intrusions, and the test set was gathered from real *port scanning* attacks launched by Nmap. In traditional AIS based IDS, a mature antibody set is generated after negative selecting the normal traffic collected by the switch, where a match denotes an anomaly activity. The accuracy entirely depends on the normal traffic set selected to train the antibody.

We argue that this IDS is constrained by the local traffic the switch has seen, and thus will lead to high false-positive rate when new normal traffic appears (which is common nowadays as new web applications keep popping up). By using the global view of the network provided by SDN, we fuse the antibody sets trained by every switch in the network as well as the intrusions recorded into a global antibody set, and distribute it to all the switches. This is equivalent to that antibody set being trained by the traffic flow over the entire network. With regards to these two metrics, false-positive and true-negative rates, BIDS is able to outperform the previous work on AIS based IDS and IDS using other mechinism for SDN with less cpu usage.

For future work, we are planning to improve the AIS algorithm by finding a more efficient presentation to describe the traffic pattern. Secondly, we will gather more normal and abnormal training datasets, to achieve better performance of the system.

ACKNOWLEDGMENTS

This research has been supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) projects EP/N033957/1, and EP/P004024/1; by the European Cooperation in Science and Technology (COST) Action CA 15127: RECODIS – Resilient communication and services; by the EU H2020 GNFUV Project RAWFIE-OC2-EXPSCI (Grant No. 645220), under the EC FIRE+ initiative; and by the Huawei Innovation Research Program (Grant No. 300952).

REFERENCES

[1] S.A. Hofmeyr, and S. Forrest. "Architecture for an artificial immune system." *Evolutionary computation, vol.8, no.4, pp.443–473,* 2000.

[2] J. Kim, and P. J. Bentley. "An evaluation of negative selection in an artificial immune system for network intrusion detection." In *Proc. the 3rd Annual Conference on Genetic and Evolutionary Computation, 2001.*

[3] F. Hosseinpour, et al. "Distributed agent based model for intrusion detection system based on artificial immune system." *International J. Digit. Cont. Tec. Appl., vol.7, no.9, pp.206,* 2013.

[4] S. Forrest, et al. "Self-nonself discrimination in a computer." Research in Security and Privacy, In *Proc. IEEE Comput. Soc. Sympos.,* 1994.

[5] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks, vol.7, no.62, pp.122–36,* Apr. 2014.

[6] W. Wang, W. He, and J. Su, "Network intrusion detection and prevention middlebox management in SDN," In *Proc. IEEE IPCCC, Dec. 2015, pp.1–8.*

[7] N. G. Kabasele, and R. Sadre, "A Two-level Intrusion Detection System for Industrial Control System Networks using P4," In *Proc. ICS-CSR 2018.*

[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Comput. Comm. Rev., vol.38, no.2, pp.69–74,* 31st, Mar. 2008.

[9] J.O. Kephart, "A biologically inspired immune system for computers," In *Proc. 4th international workshop on the synthesis and simulation of living systems, 1994, pp.130–139.*

[10] M. V. Mahoney, and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," In *Proc. International Workshop on Recent Advances in Intrusion Detection, Springer, Sep. 2003, Berlin, Heidelberg,* pp.220–237.

[11] M. A. Lopez, and O.C.M. Duarte, "Providing elasticity to intrusion detection systems in virtualised software defined networks," In *Proc. IEEE ICC, June 2015*, pp.7120–7125.

[12] S. Mehdi, J. Khalid, and S. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. 14th RAID, 2011,* pp. 161–180.

[13] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, and M. Ghogho, 2016, October. "Deep learning approach for network intrusion detection in software defined networking," In *Proc. IEEE WINCOM, Oct. 2016, pp. 258–263.*