Multimetric Online Intrusion Detection in Software-Defined Wireless Sensor Networks

Gustavo a Nunez Segura, Arsenia Chorti, Cintia Borges Margi

Multimetric Online Intrusion Detection in Software-Defined Wireless Sensor Networks

Gustavo A. Nunez Segura*, Arsenia Chorti‡, and Cintia Borges Margi*
*Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
‡ETIS UMR8051, CY Université, ENSEA, CNRS, F-95000, Cergy, France
Email: {gustavoalonso.nunez, cintia}@usp.br, arsenia.chorti@ensea.fr

***Abstract*—Software-defined wireless sensor networks have attracted considerable attention in recent years as they simplify the network management and provide the framework to automate infrastructure sharing. On the other hand, the centralization and planes' separation can turn SDNs vulnerable to new types of denial of service attacks. Existing intrusion detection approaches are not in general suitable for restricted networks or do not achieve optimal detection rates. This work aims at fulfilling both requirements by using a new lightweight, multimetric, online change point detector to monitor performance metrics that are impacted when the network is under attack. There are two major novelties in the proposed detector referring to previous works: first, we move to a purely online detector, secondly, we monitor in parallel multiple metrics, increasing the detection vector space to different types of attack. Our tests show that intrusion detection monitoring control overhead and data packets delivery rate in a SDWSN results in enhanced detection rates over 96% in all topologies and levels of attacks. We finally show that with a high probability (exceeding 89% in all cases) it is possible to identify the "type" of the attack.**

***Index Terms*—Intrusion detection, software-defined networking, wireless sensor networks**

I. Introduction

Software-defined wireless sensor networks (SDWSN) emerged from the application of the software-defined networking (SDN) paradigm in wireless sensor networks (WSN). The objective was to use SDN characteristics to leverage solutions for WSN challenges, in particular concerning flexibility and resource reuse [1]. Conversely, the SDN centralization and the planes' separation turn the network vulnerable to new security threats, mainly by those attacks targeting the SDN controller [2], [3].

SDWSNs inherit the vulnerabilities of SDNs, and, moreover, they are prone to novel types of attacks that are specific to SDWSNs and are related to their constraints in terms of resources. An example, the low storage capacity of SDWSN devices limits the memory that can be assigned for flow tables and buffers. This constraint turns the network prone to saturation attacks. Also, SDWSN nodes typically operate at low rates and with limited processing power. This means that a saturation attack can also result in a denial of service (DoS) attack, e.g., the gateway between the SDN controller and the WSN has a radio module of limited bandwidth, rendering it a weak link even when the controller has enough resources to overcome an attack [1].

One challenge for SDWSN security is to fine-tune SDN security solutions accounting for the resource constraints of SDWSNs, a topic that has already attracted considerable attention in the literature in Internet of things (IoT) and WSN networks using SDNs. However, current security proposals for restricted SDN-based networks show high attack detection rates (over 90%) only in small topologies, implemented and tested in environments not suitable for restricted networks (e.g., using OpenFlow). Moreover, these proposals lack mechanisms for identifying the type of the attack, which are necessary to implement mitigation strategies.

To fill this gap, the main contribution of this work is the proposal of an intrusion / DDoS attack detector that is suitable for constrained SDWSNs, achieving very high detection rates, exceeding in all of the investigated scenarios 96% and in some of them (the most intense attacks) reaching 100%. A second important advantage of our solution is that it provides a probabilistic identification of the "type" of the attack. To reach these targets, we have built a multimetric online detector, monitoring in parallel both the data packets delivery rate and the control packets overhead, using two independent change point (CP) detectors of $O(N)$ complexity, where N is the length of the observed data, which can be as sort as $N = 200$ samples. Furthermore, we increased the detector output space by providing multiple options between the detection rate and the speed of detection. Our results illustrate the trade-off between the detection accuracy and the agility of the detector.

II. Related work

In [4], an SDN-based IoT framework with security support was proposed, developed for OpenFlow – which limits its use in networks composed of constrained nodes – while results were presented in networks with only five nodes. In [5], a distributed DoS (DDoS) detector was proposed, monitoring the difference of packets received by the controller using *cosine similarity*. This proposal is based only on the controller view, which could limit the types of attack it can detect. In [6], a SDWSN trust management and routing mechanism was proposed. The focus of this contribution was on identifying selective forwarding attacks and new flow requests. Their results showed attack detection rates between 80%-90% when 10% of nodes were attackers. In [7], a multilayer security framework using authentication and intrusion detection based on the difference of the residual energy of the nodes was developed. Their multilayer approach detects different types of attacks but at the cost of a reduced in detection rate (around 87%).

The principal shortcoming of these previous works is that high detection rates, close to 100%, were only achieved in

978-1-7281-8903-1/20/$31.00 ©2020 IEEE

very small networks using OpenFlow, which might not be suitable for restricted networks. A second shortcoming is the lack of mechanisms to identify the type of the attack. This information is paramount to implement mitigation strategies. The present study builds upon our previous works in [8] and [9]. In the first work [8] we analyzed the impact of different types of attacks on various performance metrics and identified the data packet delivery and the control overhead rates as the most impacted. In [9] on the other hand, we proposed a universal CP DDoS detector. In the present, we move to an entirely online multimetric CP detector, which uses two CP detectors independently optimized for different types of attack. This strategy allow us to obtain high detection rates for different attacks in topologies up to 100 nodes, and, more importantly, to identify the type of the attack we are detecting. In addition, our proposal is suitable for restricted devices because the processing overhead lays on the controller

III. Proposed Online Detector

Our previous work [8] provided valuable insight regarding the impact of (i) the false data flow forwarding attack, dubbed as FDFF, in which a malicious node sends data packets to its neighbors with unknown flows, and, (ii) false neighbor information attack, dubbed as FNI, in which a malicious node modifies neighborhood report packets in their route before reaching the controller. It was shown that FDFF attacks induce substantial changes in mean control packet rates while FNI attacks induce important changes in mean data packets delivery rates. Building on this analysis, we formulate the attack detection problem as a hypothesis test, examining whether a change has occurred in the mean value of the time series observed for these two metrics. Next, we present the online CP algorithm that was implemented.

We employ a cumulative sum (CUSUM) based algorithm for the detection of a change in the mean value of a time series, this choice allows us to alleviate the need for any parametric model with respect to the impact of the attack [9]. There are two major novelties in the proposed detector in this work with respect to the one presented in [9]: first, we move to a purely online detector, unlike [9] in which a hybrid offline-online algorithm was presented; secondly, we monitor in parallel multiple metrics, increasing the detection vector space to different types of attack and provide a probabilistic identification of the type of the attack, which to the best of our knowledge breaks new ground.

We begin with a short description of the online CP algorithm. Let $\{X_n : n \in \mathbb{N}\}$ be the time series of the metric monitored. Using Wold's theorem we can assume that, for $X_1, ..., X_N$, each sample can be expressed as $X_n = \mu_n + Y_n$, where $\{\mu_n, n \in \mathbb{N}\}$ is the mean of the time series and $\{Y_n : n \in \mathbb{N}\}$ is a random zero mean term, so that we can rewrite X_n as:

$$X_n = \begin{cases} \mu + Y_n, & n = 1, \dots, m + k^* - 1 \\ \mu + Y_n + I, & n = m + k^*, \dots \end{cases} \tag{1}$$

where $k^* \in \mathbb{N}^*$ represents the unknown time of change and μ, $\mu + I \in \mathbb{R}$ represent the mean values before and after k^*, respectively. In the present we assume a period of no change in the mean of at least m samples, i.e., during the first m samples of our observation there is no change so that $\mu_1 = \ldots = \mu_m$. During this period our detector "learns" in real-time the statistics of the observed time series, and, the mean value in particular. We note that m can be kept small, as discussed in Section IV. Finally, the statistical hypothesis test is articulated as,

$$\begin{aligned} H_0 &: I = 0 \\ H_1 &: I \neq 0. \end{aligned} \tag{2}$$

The online CP detector is a stopping time stochastic process defined as:

$$\tau(m) = \begin{cases} \min\{l \in \mathbb{N} : |TS(m,l)| \geqslant F(m,l)\} \\ \infty, \text{otherwise} \end{cases}. \tag{3}$$

where $TS(m,l)$ is the detector statistic, $F(m,l)$ is the detection threshold defined below and l is the monitoring window (i.e., the window in which we perform the hypothesis test). The CP detector has the following properties:

$$\lim_{m\to\infty} P\{\tau_m < \infty | H_0\} = 1 - \alpha, \tag{4}$$

ensuring that the probability of false alarm is asymptotically bounded by $1 - \alpha \in (0, 1)$, and,

$$\lim_{m\to\infty} P\{\tau_m < \infty | H_1\} = 1, \tag{5}$$

ensuring that under H_1 the asymptotic power is unity. Under these conditions, $F(m,l)$ is defined as,

$$F(m,l) = c_a g_\gamma(m,l), \tag{6}$$

where c_a is the critical value determined and g_γ is the weight function defined as:

$$g_\gamma(m,l) = \sqrt{m}\left(1 + \frac{l}{m}\right)\left(\frac{l}{l+m}\right)^\gamma, \tag{7}$$

where the sensitivity parameter $\gamma \in [0, 1/2)$. The online algorithm uses the standard CUSUM detector, given by:

$$\Gamma(m,l) = \frac{1}{\widehat{\omega}_m}\left(\sum_{i=m+1}^{m+l} X_i - \frac{l}{m}\sum_{i=1}^{m} X_i\right) \tag{8}$$

with $\widehat{\omega}_m$ denoting the asymptotic variance, that captures the serial dependence between observations.

The corresponding threshold is $F^\Gamma(m,l) = c_a^\Gamma g_\gamma(m,l)$ and the critical value are defined as:

$$\begin{aligned} \lim_{m\to\infty} P\{\tau_m < \infty\} &= \lim_{m\to\infty} P\left\{\frac{1}{\widehat{\omega}_m} \sup_{1\leqslant l\leqslant\infty} \frac{|\Gamma(m,l)|}{g_\gamma(m,l)} > c_a^\Gamma\right\} \\ &= 1 - \alpha. \end{aligned} \tag{9}$$

Summarizing, the overall algorithm has 3 main steps:

- Step 1: define the values of the quantities m, γ and the confidence level α and set l.
- Step 2: after collecting m samples of the metric, $\Gamma(m,l)$ (8) and the weight function in (7) are calculated for every l on the monitoring period to then apply (9).
- If a CP is detected, the online process stops. Conversely, if the period l ends, a new monitoring period is defined.

IV. Experimental Setup

We generated a dataset comprising simulations of 240 FNI attacks and 240 FDFF attacks and recorded the data packet delivery and control overhead rate time series. The full set of time series were then split in two groups; one group for training, in order to determine the optimal values of $\{m, \gamma\}$ for each type of attack and each observed metric and the other group for validation. In terms of topology, all simulations were performed on square grids with either 36 or 100 nodes. For each topology, we varied the number of intruders (attackers) in three proportions: 5%, 10% and 20% of the total of nodes in the network.

We employed the CP algorithm presented in Section III to detect changes in the delivery rate and the control packets overhead mean rates of an SDWSN under attack, by using two CP detectors running in parallel and each one of them monitoring one of the two metrics. First, we executed the algorithm on the training group varying $m \in \{100, 150, 200\}$ and $\gamma \in \{0, 0.15, 0.25, 0.35, 0.45, 0.49\}$. The objective was to determine the combination of parameters that could provide the best detection performance for each type of attack (FDFF or FNI types) regardless of the number of intruders. For the FDFF attack we identified the optimal $\{m, \gamma\}$ to detect changes in the control overhead time series, while in the FNI attack we identified the optimal $\{m, \gamma\}$ to identify CP in the data packets delivery rate time series.

Furthermore, we added flexibility in the detection by introducing a "detection score" metric to capture the relative importance that is given to the detection rate (DR) versus the detection speed (captured by the median of samples required to detect the attack). The DR is defined as the ratio of successfully detected attacks over the total number of attacks. The detection time median, DTM, is the median of the number of samples required to detect the attack. The proposed detection score metric, P_{DS}, is defined as:

$$P_{DS}(A, B) = A(1 - S) + B(DR), \qquad A + B = 1, \quad (10)$$

where $S = \frac{DTM}{l}$ and l is the number of samples monitored after the attack starts. A and B are constants determining the relative weight of each term. In our tests we used 5 combinations, $(A, B) \in \{(1, 0), (0.8, 0.2), (0.5, 0.5), (0.2, 0.8), (0, 1)\}$ and recorded the results when giving more weight to the speed of detection ($A > B$) versus the detection rate ($A < B$).

During validation, we executed in parallel two CP algorithms, the first one monitoring the control packets overhead and the second one the data packets delivery rates of the validation set. The validation set comprises both FDFF and FNI attack simulations, all possible topologies and attack intensity levels, with FDFF attacks mounting for 50% of the whole set. The two parallel CP detectors were optimized separately during the training, i.e., in the validation we used the optimal pairs (m, γ) identified for each (A, B) in order to maximize the $P_{DS}(A, B)$ under either an FDFF or an FNI attack. Whenever either CP was triggered, we stopped the detection, declared an attack, and, also, made a guess regarding the type of the attack. If the CP monitoring the control overhead was triggered first, we declared a FDFF type of attack, alternatively, if the CP monitoring the data packet delivery rate was triggered first we declared a FNI type of attack.

The SDWSN implementation used IT-SDN, without changing the default configuration [10], and the simulations were performed using the COOJA simulator [11], emulating sky motes. The sensor nodes were programmed to transmit one data packet every 30 seconds and one management packet every 2 minutes, both with payload of 10 bytes. The data packets contained the application information and the management packets contained the information required by the network management plane [12].

The data packets delivery rate and the control packets overhead were observed every two minutes, considering the exchange of messages in the whole network during this window of time. The delivery rate was calculated by dividing the number of data packets successfully received by the number of data packets sent. The control packets overhead was quantified as the number of control packets sent. Since we took samples every two minutes, we decided to run each single simulation for 10 hours. During the first 8 hours the network operated normally (i.e., for 240 samples there was no change), then the attack was triggered. This imposed a bound $m < 240$.

V. Results and Analysis

Our analysis is separated according to the dataset groups. In Section V-A we present and analyze the result from the training dataset. In Section V-B we show and analyze the results obtained from the validation dataset.

A. Training dataset analysis

We grouped the results in three passes regarding α, the attack type, and the metric monitored. We investigated scenarios with $\alpha \in \{0.90, 0.95, 0.99\}$. For each α, we split the results by attack type and then sorted them according to the metric monitored. We calculated the average detection score P_{DS} for all topologies, attack scenarios and combinations of m and γ. An interesting result was that in 90% of all cases P_{DS} was maximized when using $m = 200$, making this a universally optimal choice.

The next step was to choose the values of γ that maximized P_{DS} for each monitoring metric and attack. The results in [9] suggested that, in terms of detection rate and samples required to detect the attack, the detection performance of FNI attacks was better when monitoring the data packets delivery rate than when monitoring the control overhead. Conversely, the detection performance of FDFF attacks was better when monitoring the control overhead. From this, we decided that γ should be chosen for each CP detector separately to maximize the detection score.

Fig. 1 shows the average value of P_{DS} as a function of γ and α for the case of the FDFF attack. In Fig. 1a we show that for $A = 1$ (i.e., prioritizing faster detection) the higher results of P_{DS} were obtained for $\gamma = \{0.35, 0.45\}$. In the case where $A = B = 0.5$, P_{DS} was maximized when $\gamma = \{0.25, 0.35\}$. And lastly, for $B = 1$ (i.e., prioritizing the detection rate), the best results were obtained for $\gamma = \{0, 0.15, 0.25\}$. What we observed is that there is a clear relation between the value of γ and the detection performance; using higher values of γ we could expect a

TABLE I: γ that maximizes P_{DS}

P_{DS}	γ		
	0.1	**0.05**	**0.01**
Best γ for control overhead CP detector			
$A = 1$ and $B = 0$	0.45	0.45	0.45
$A = 0.8$ and $B = 0.2$	0.35	0.35	0.45
$A = 0.5$ and $B = 0.5$	0.25	0.35	0.45
$A = 0.2$ and $B = 0.8$	0.25	0.25	0.35
$A = 0$ and $B = 1$	0	0	0
Best γ for delivery rate CP detector			
$A = 1$ and $B = 0$	0.45	0.45	0.45
$A = 0.8$ and $B = 0.2$	0	0.15	0.15
$A = 0.5$ and $B = 0.5$	0	0	0.15
$A = 0.2$ and $B = 0.8$	0	0	0
$A = 0$ and $B = 1$	0	0	0

faster response while using lower values we could expect higher accuracy in the detection.

Then, Fig. 2 shows of the metric P_{DS} for the FNI attack. In this case, the results were not as clear-cut as we observed for the FDFF attack. However, as shown in Fig. 2a, we could expect faster detection when using higher values of γ; and as shown in Fig. 2c, we could expect more accurate detection when using lower values in γ. The main difference with respect to the results obtained for the FDFF attack was when calculating P_{DS} for $A = B = 0.5$. In this case the results showed a behavior similar to the behavior when $B = 1$. One reason for this, and the one for which we have more evidence, was that the DTM behavior when detecting FNI attack was flatter than when detecting FDFF attack, thus the changes in the detection rate had more influence in the metric P_{DS}. We can see this comparing Figures 1a and 2a on their extreme values. In Table I we show the values of γ that maximize P_{DS}. In cases where more than one values provided the same or very comparable results, we chose one of them arbitrarily.

B. Validation dataset results analysis

In this subsection we analyze the performance of the CP detectors running on the second group of the dataset, using the values of m and γ that maximize P_{DS}. For this analysis, we set two detectors running simultaneously, one detector monitoring the control overhead and the other monitoring the data packets delivery rate. This gave us the opportunity to test the hypothesis in [9] about the possibility of identifying the type of the attack based on the first detector triggered.

To test this, we counted all the events where the attacks were correctly detected and classified them according to the CP detector that was triggered first. From these results, we calculated the probability of the attacks being detected first by the control overhead CP detector or by the data packets delivery rate CP detector. As shown in Fig. 3, in the case of the FDFF attack, the probability of the detector monitoring the control overhead is triggered first happens with a probability between 89% and 98%. In case of the FNI attack, the probability that the detector monitoring the data packets delivery rate triggers the alarm first is interestingly 100%. These results show that there is evidence to prove the conjecture drawn up in our previous work [9] about the relation metric / attack.

Fig. 5 depicts DR and $1-S$ when the network was under FDFF attack. We observed that in terms of detection rate (Fig. 5a), for four out of five configurations we achieved $DR > \alpha$, and, only for $A = 1$ the DR did not reach the target α. Then, in terms of $1 - S$, the results showed that we obtained the fastest detection when $A = 1$, as intuitively expected by inspecting Figs. 5 and 6. Considering both DR and S, the results for $A = 0.8$ provided the best trade off.

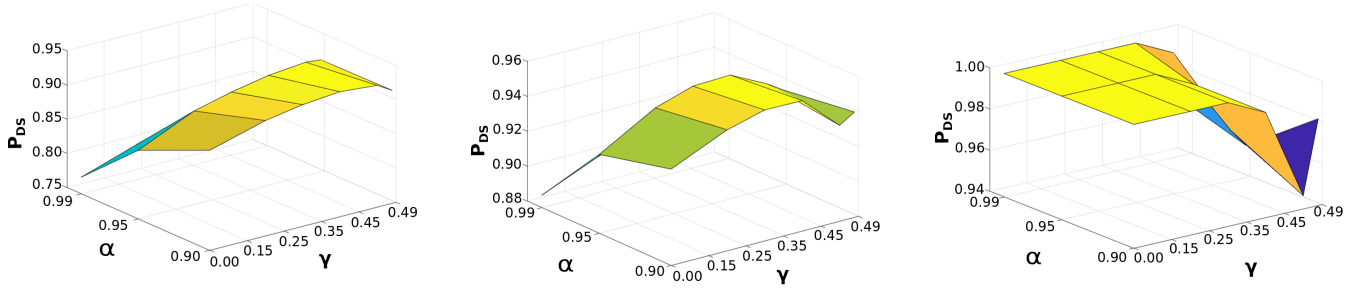
In the case of FNI attack, in Fig. 6 the DR observed was not as good as the results obtained in the FDFF attack. For the case of $\alpha = 0.95$ only for two configurations $DR > \alpha$, for $A = 0$ and $A = 0.2$, while for $\alpha = 0.99$ none of them achieved it. The difference in the detection rate for FDFF and FNI attacks was already noticed in the training analysis. Comparing Figs. 1c and 2c, the values of P_{DS} when $A = 0$ (i.e., considering only the detection rate) have a difference around 0.04 for $\gamma = 0$ and around 0.10 for $\gamma = 0.49$. In terms of the time required to detect the attack, both obtained similar results. Thus, considering both detection rate and time required to detect the attack, when $A = 0$ and $A = 0.2$ the detection rate was equal or over α, for $\alpha = \{0.90, 0.95\}$, and $1 - S$ was higher than 0.87. Diversely, when $A = 1$ the metric $1 - S$ obtained the highest results (i.e., the fastest detection) but the DR dropped below 0.90.

Finally, we analyzed the detection performance irrespective of the type of the attack. Our objective was to simulate a real scenario in which both CP detectors run simultaneously in a network prone to both attacks. In Figs. 7a and 7b we show DR and $1 - S$, respectively. Our results show that the proposed multimetric online intrusion detector was able to reach a detection rate equal or over α when $A = \{0, 0.2\}$. This means, to overall guarantee a high DR the detection score should be evaluated with $A > B$. If we compare the value of $1 - S$ for $A = 0$ and $A = 1$, the highest difference is about 0.05 when $\alpha = 0.90$, which translates to a lag of 3 samples in the time series (see Section IV). This means, if we choose to configure the algorithm to maximize DR, the detector is triggered 3 samples later on average. Conversely, if we choose to configure the algorithm for fastest detection, the DR can fall below 90%.

Summarizing, we chose the pairs of (m, γ) that maximized the detection score P_{DS}. Our results showed that the metric P_{DS} was maximized when setting $m = 200$ while the optimal value of γ depended on the relative weight given to A and B. When using $\gamma = \{0.45, 0.49\}$ we reduced the time to detect the attack but this had an adverse effect on the detection rate. On the other hand, when $\gamma = \{0, 0.15\}$ the detection rate and the time to detect the attack increased. Then, we showed that there is a probability over 0.89 that in case of a FDFF attack, the control overhead CP detector will be triggered first. Conversely, in the case of a FNI attack, the delivery rate CP detector always raised the alarm first. We were able to detect the attack with $DR > \alpha$ when $B > A$. On the other hand, the fastest detection was obtained when $A = 1$ but with a detection rate equal or less than 0.93.

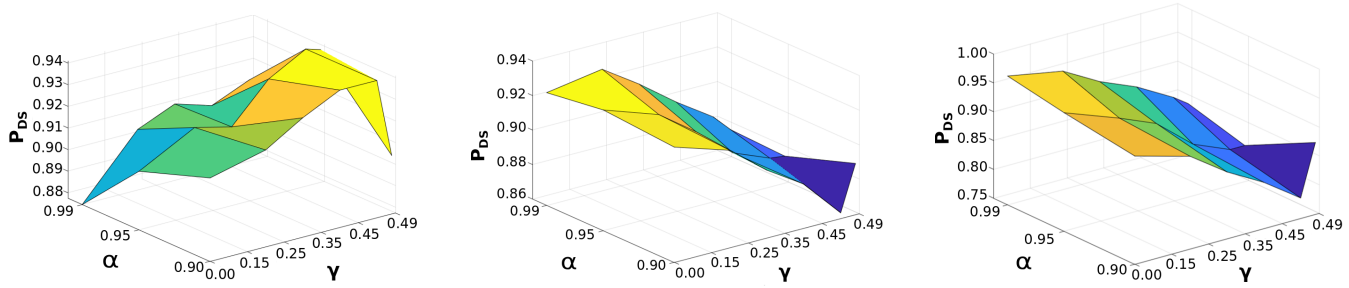
VI. Conclusion

We proposed a novel multimetric online intrusion detection algorithm for SDWSNs. There are two major novelties in



(a) $A = 1$ and $B = 0$ (b) $A = B = 0.5$ (c) $A = 0$ and $B = 1$

Fig. 1: Metric P_{DS} for FDFF attack



(a) $A = 1$ and $B = 0$ (b) $A = B = 0.5$ (c) $A = 0$ and $B = 1$

Fig. 2: Metric P_{DS} for FNI attack

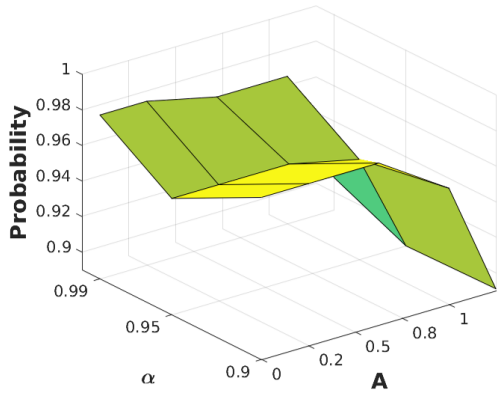


Fig. 3: Probability of control overhead CP detector being triggered first in case of FDFF attack

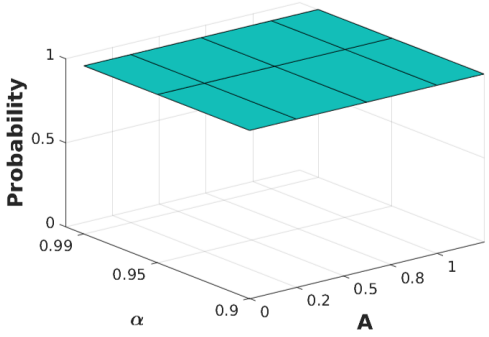


Fig. 4: Probability of delivery rate CP detector being triggered first in case of FNI attack

the proposed detector regarding previous works: first, we move to a purely online detector, secondly, we monitor in parallel multiple metrics, increasing the detection vector space to different types of attack. The proposal was tested for two DDoS attacks simulating SDWSNs of 36 and 100 nodes monitoring their data packets delivery rate and control overhead.

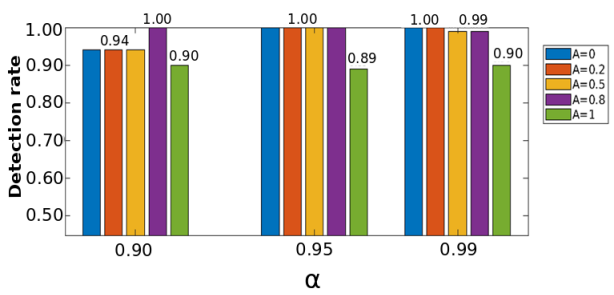
First, we executed the algorithm on a training dataset to obtain the values of the parameters that maximized the performance based on a detection score that captured both the detection rate and the time required to detect the attack. The results showed performance was maximized when $m = 200$. Then, we varied the CP sensitivity parameter (γ) varying the weight we gave to the detection rate and to the time to detect the attack.

We tested the algorithm in a second dataset using the values of m and γ chosen. We showed that there is a probability over 0.89 that in case of a FDFF attack, the control overhead CP detector will be triggered first. On the other hand, the delivery rate CP detector was triggered first in all the cases the network was under a FNI attack. In terms of detection performance the detection rate was between 0.96 and 0.99.

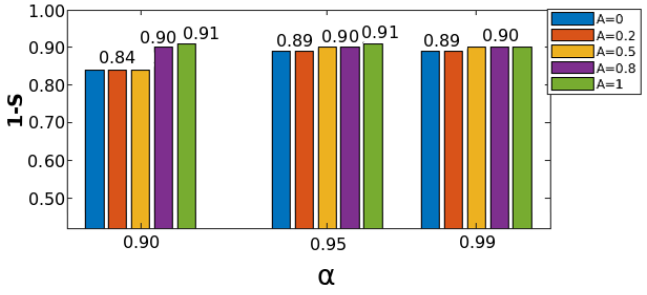
We were able to reach very high detection rates and also provide with high probability a good discriminator of the type of the attack. In future work, we would like to propose a decentralised version of the multimetric intrusion detector to obtain information about the attackers' location.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by the ELIOT project (ANR-18-CE40-0030 / FAPESP 2018/12579-7). Gustavo A. Nunez Segura is supported by Universidad de Costa Rica.

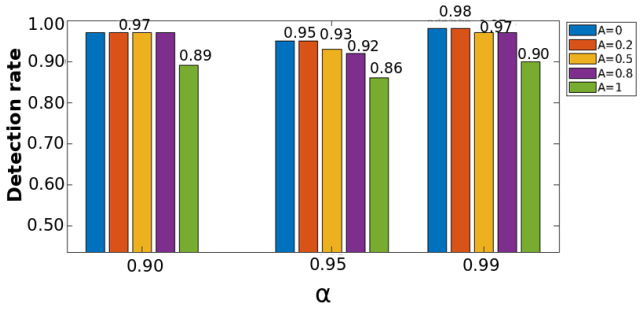


(a) Detection rate

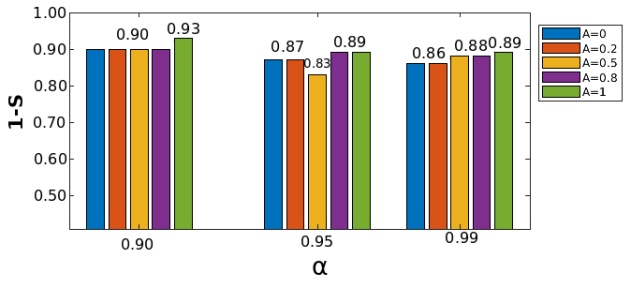


(b) 1-S

Fig. 5: Detection performance of FDFF attack

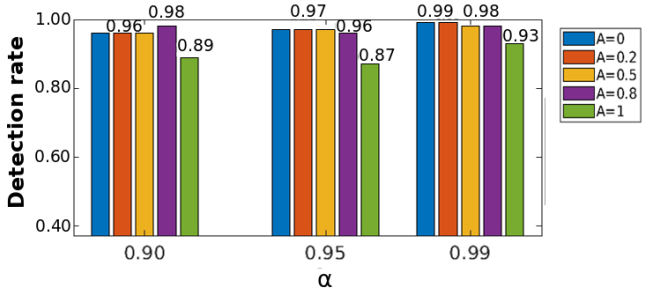


(a) Detection rate

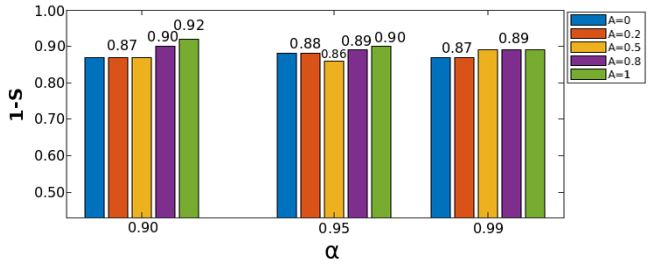


(b) 1-S

Fig. 6: Detection performance of FNI attack



(a) Detection rate



(b) 1-S

Fig. 7: Detection performance regardless the type of the attack

REFERENCES

[1] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[2] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, Fourthquarter 2015.

[3] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in Software-Defined Networking: Threats and Countermeasures," *Mobile Netw. and Appl.*, vol. 21, no. 5, pp. 764–776, Oct 2016.

[4] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *27th Int. Telecommun. Netw. and Appl. Conf. (ITNAC)*, Nov 2017, pp. 1–6.

[5] D. Yin, L. Zhang, and K. Yang, "A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework," *IEEE Access*, vol. 6, pp. 24 694–24 705, 2018.

[6] R. Wang, Z. Zhang, Z. Zhang, and Z. Jia, "ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs," *Computer Networks*, vol. 139, pp. 119 – 135, 2018.

[7] C. Miranda, G. Kaddoum, E. Bou-Harb, S. Garg, and K. Kaur, "A collaborative security framework for software-defined wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2020.

[8] G. A. N. Segura, C. B. Margi, and A. Chorti, "Understanding the Performance of Software Defined Wireless Sensor Networks Under Denial of Service Attack," *Open Journal of Internet Of Things (OJIOT)*, 2019, special Issue: Proc. Int. Workshop Very Large Internet of Things (VLIoT 2019) in conjunction with the VLDB 2019 Conf. Los Angeles, United States.

[9] N. S. Gustavo, S. Skaperas, A. Chorti, L. Mamatas, and B. M. Cintia, "Denial of Service Attacks Detection in Software-Defined Wireless Sensor Networks," in *SecSDN IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020.

[10] R. C. A. Alves, D. A. G. Oliveira, G. A. Nunez Segura, and C. B. Margi, "The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks," *IEEE Access*, vol. 7, pp. 115 093–115 108, Aug 2019.

[11] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Proc. IEEE Conf. Local Comput. Netw. (LCN)*, Nov 2006, pp. 641–648.

[12] T. Luz, G. Nunez, C. Margi, and F. Verdi, "In-network performance measurements for Software Defined Wireless Sensor Networks," in *16th IEEE Int. Conf. Netw., Sens. and Control (ICNSC 2019)*, May 2019.