Bio-inspired AI for Network Intrusion Detection on Industrial IoT

Zachary Wu[*]
Department of ECE
University of Waterloo
zachary.wu@uwaterloo.ca

Yen Zein Kok[*]
Centre for Computational Mathematics
University of Waterloo
ykok@uwaterloo.ca

Madhav Malhotra[*]
Department of ECE
University of Waterloo
madhav.malhotra@uwaterloo.ca

Akira Yoshiyama[*]
Department of ECE
University of Waterloo
akira.yoshiyama@uwaterloo.ca

Abstract*—The number of deployed Internet of Things (IoT) devices has grown to over 10 billion by 2024. This includes Industrial IoT (IIoT) devices deployed in critical infrastructure like water treatment, energy distribution, and food processing facilities. Unfortunately, IIoT devices are often the least secure devices in an organisation. Given their computational limits, they cannot accommodate modern cyberdefences like AI-enabled extended detection & response (XDR). This paper explores the use of a bio-inspired negative selection algorithm that is lightweight enough to run on IIoT devices. It finds that such algorithms offer no advantages over shallow ML.** **Keywords—*IoT, AI, cybersecurity, intrusion detection.

I. Introduction

A. The IoT Threat Landscape

IoT devices are an increasingly vital part of the modern economy. In the past decade, billions of these devices have been deployed in residential, commercial, and industrial settings [1]. Industrial IoT (IIoT) devices, in particular, have achieved widespread economic impact by optimising industrial operations through large-scale data collection [2]. This includes usage in critical infrastructure like water treatment and energy distribution facilities.

Unfortunately, many cybersecurity vulnerabilities exist for common IIoT devices. These may be physical attacks to block radio-frequency communication or network attacks that disclose confidential data to unauthorised parties [3]. At the same time, critical infrastructure like water treatment facilities [4] and fuel distribution networks [5] are being targeted in increasingly damaging cyberattacks.

Traditional cyberdefences based on human expertise and rules-based software fail to scale to modern cyberattacks which rapidly evolve [6]. This is why an increasing number of organisations are deploying cyberdefences enabled by Artificial Intelligence (AI) to assist in tasks like detecting malware, detecting intruders on private networks, and classifying anomalous insider activities [7]. Still, modern AI algorithms have found relatively little adoption in IoT devices, since common deep learning (DL) algorithms are often too computationally intensive to run on these devices.

B. Bio-inspired AI and Cybersecurity

Bio-inspired AI algorithms are a category of algorithms which learn to solve optimisation problems using techniques which mimic naturally intelligent behaviour [8].

By analysing defensive behaviours in natural systems like the immune system, researchers have been inspired to create algorithms which mimic these behaviours for cybersecurity applications [9]. For instance, artificial immune systems [10], Negative Selection Algorithms (NSA) [11], and Positive Selection Algorithms (PSA) [12] have been created for tasks like detecting malware or network intruders. In brief, NSA and PSA find patterns in statistics to identify benign or malicious network activity. They compare ongoing network activity against these prior patterns [12].

Our work proposes the use of NSA and PSA to create computationally inexpensive network intrusion detection algorithms which can run on common IoT devices. We compare the performance and computational cost of these algorithms to shallow machine learning (ML) methods like random forests and logistic regression. The rest of this paper is structured as follows; Section II describes related works, Section III explains the algorithms tested, Section IV details our methodology, Section V notes our experimental results, and Section VI concludes our work with a discussion.

II. Related Works

Many papers have been published related to ML, IoT, and bio-inspired AI. Though few papers have considered the nexus of these domains. The most abundant research topic is on ML systems for intrusion detection. For instance, [13] shows the application of deep recurrent neural networks to protect industrial control systems from Distributed Denial of Service (DDoS) attacks. The work shows that DL systems are performant in environments where compute is not restrictive, even enabling multiple defence systems to run in real-time for specific cyberattack types.

[*] Authors listed in reverse alphabetical order

In contrast, [14] shows the limitations of compute intensive algorithms in environments of distributed and mobile nodes, such as vehicular networks. It demonstrates the complexity of controller-peripheral coordination when offloading computational jobs to cloud environments, which was needed to enable a clustering algorithm on large data volumes. Although possible, the scheme adds constraints to the networking capabilities of IoT devices, which would preferably be avoided by lightweight ML algorithms.

Unfortunately, there is a relative paucity of research on this latter topic. Most historical works on cybersecurity for IoT devices rely on statistical methods [15] or historical signatures [16] to detect incoming cyberattacks. However, the rapid evolution of cyberattack methods changes the data distributions statistical models and human-engineered signatures were trained on. This inhibits the long-term efficacy of these solutions in practice [17].

Furthermore, research on lightweight ML techniques to bridge this gap has been plagued by issues caused by methodological pitfalls [17] and poor datasets. Studies such as [18] have relied on recording a custom, and private, dataset for a single IoT device in simulated lab conditions. This makes it difficult to assess the generalisability of the reported academic results to real-world contexts. Still, there are some common public datasets in other academic research [19], like the NSL-KDD [20], CIC-IDS2017 [21], and CIDDS-001 [22]. Sadly, these have issues like being too old to include traffic from modern IoT devices [20], relying on simulated traffic instead of real devices [22], and featuring large class imbalances [21].

As seen, solutions have been proposed to address the important need for IoT devices with several types of methods: statistical, signature-based, shallow ML, DL, bio-inspired, and more. Still, methodological flaws and a lack of generalisation continue to be large challenges.

III. Our Proposed Algorithms

To contribute to this field, our work aims to use modern datasets and a lightweight algorithm. Specifically, we implement a modified version of NSA and PSA to enable lightweight intrusion detection.

The standard implementations of NSA and PSA are described in [11] and [12] respectively. The algorithms take inspiration from the immune system, where immune cells like T-cells distinguish foreign antigens from native body cells via proteins on the cells' surface [12]. Analogous to this, NSA and PSA recognise various classes by creating 'detectors,' which are the typical data features of some class. The similarity of these detectors can be compared to the features of unclassified data to make predictions.

In our binary classification use case, these algorithms train on an input dataset $D \in \mathbb{R}_{n\times(m+1)}$ with n records, each with $m \in \mathbb{N}$ features and a binary label. Using an arbitrary optimisation algorithm, $N_D \in \mathbb{N}$ detectors are chosen. Each detector $\overrightarrow{d} \in \mathbb{R}^m$ has m features that are optimised to be as close as possible to the features of benign data (negative selection) or malicious data (positive selection).

For an incoming network request $\overrightarrow{x} \in \mathbb{R}^m$, traditional NSA and PSA implementations compare the Euclidean distance between the network request and the ith detector:

$$\left|\left|\overrightarrow{x} - \overrightarrow{d_i}\right|\right|_2$$

If this distance is less than some threshold $\tau \in \mathbb{R}$ for the ith detector, the network request is classified as benign (NSA) or malicious (PSA). In our work, however, we replace the Euclidean distance with Manhattan distance. This reduces the computational cost of inference and adds support for microcontrollers without floating point units.

We compare the performance and computational cost of NSA and PSA with shallow ML algorithms, which we further detail in the methodology.

IV. Methodology

All models were trained on containers with access to a 4-core Intel Xeon CPU @ 2.20 GHz, with 30 GB of RAM and 73 GB of disk space. No GPU accelerators were used during training. Hyperparameter search was done using the Hyperopt library [23], which includes the TPE optimization algorithm, to a maximum of 100 trials. The detailed hyperparameter search space for each model can be found in the supplementary materials.

We conducted a hyperparameter search each time we trained a model on a dataset, using 1% of the data (without overlap on the training or testing sets) as a validation set.

We trained and evaluated each model on 3 subsets of the original CICIoT2023 dataset to assess each algorithm's improvement with increased data. The datasets were constructed by selecting a proportion of rows (1, 5, or 10%) uniformly at random, stratified by attack class. We filtered out the "Protocol Type" feature for all datasets due to its categorical nature and redundancy with other features. For the 1% and 5% datasets, we only included the top 5 and 24 features as determined by feature importance on a trained Random Forest model (see supplementary materials).

TABLE I
Dataset Sizes

Dataset (% of original)	Rows	Features	Total Size (MB)
1	466,869	5	119
5	2,334,325	24	595
10	4,668,665	45	1190

Models were evaluated using PR AUC for models that supported probabilistic prediction: XGBoost, Random Forest (RF), Logistic Regression (LR), and Linear SVM. PSA and NSA used ROC Analysis. The models were exported to C and C++ code using the `m2cgen` library and compiled with `gcc` using the C11 standard and `-Os -s` flags.

Due to the limited RAM of common microcontrollers, we limited the final compiled binary size of each model to 256 KB. To this end, we limited hyperparameters that grew model size: the maximum depth and number of estimators for the

XGBoost and Random Forest models, and the number of detectors for NSA and PSA.

Computational performance metrics were evaluated using scripts shared in the supplemental materials.

V. Results

The recorded performance metrics, including precision, recall, and F1-score, are presented as macro-averages.

TABLE II
Performance Metrics

Model	Dataset	Performance Metrics				
		Accuracy	Precision	Recall	F1	PRAUC
NSA	1%	0.878	0.560	0.798	0.576	0.345
	5%	0.976	0.488	0.500	0.494	0.000
	10%	0.976	0.488	0.500	0.494	0.000
PSA	1%	0.980	0.774	0.944	0.838	0.409
	5%	0.979	0.770	0.920	0.827	0.534
	10%	0.978	0.762	0.909	0.818	0.562
LR	1%	0.969	0.712	0.926	0.780	0.707
	5%	0.956	0.674	0.975	0.747	0.691
	10%	0.952	0.665	0.975	0.735	0.646
Linear SVM	1%	0.970	0.711	0.908	0.775	0.670
	5%	0.964	0.698	0.978	0.774	0.784
	10%	0.980	0.771	0.987	0.846	0.783
Random Forest	1%	0.979	0.764	0.988	0.840	0.917
	5%	0.997	0.960	0.965	0.962	0.981
	10%	0.978	0.756	0.981	0.832	0.765
XG Boost	1%	0.994	0.923	0.943	0.936	0.950
	5%	0.996	0.953	0.970	0.962	0.978
	10%	0.996	0.953	0.971	0.961	0.976

Next, we evaluate the computational cost of each model. Specifically, the inference time (ns) is the average time to perform an inference across 1000 trials, the Unique Set Size (USS) is the RAM (kB) that would be freed by terminating the process, and bytes written or read are on the hard disk. Note that no bytes should be read/written to the hard disk as flash memory is often unavailable on microcontrollers. Up to 90% of the USS is related to the C++ runtime, not the model. This is explained in the supplementary materials.

TABLE III
Computational Performance Metrics

Model	Dataset	Computational Performance Metrics			
		Inference Time (ns)	USS (kb)	Bytes read	Bytes written
NSA	1%	3100	106496	0	0
	5%	17000	131072	0	0
	10%	31000	155648	0	0
PSA	1%	3000	105366	0	0
	5%	15000	126976	0	0
	10%	27000	151552	0	0
LR	1%	40	110592	0	0
	5%	62	106496	0	0
	10%	90	106496	0	0
Linear SVM	1%	42	106496	0	0
	5%	61	106496	0	0
	10%	88	102400	0	0
Random Forest	1%	2300	229376	0	0
	5%	2300	425984	0	0
	10%	2300	122880	0	0
XG Boost	1%	210	135168	0	0
	5%	210	147456	0	0
	10%	210	155648	0	0

VI. Discussion

There were clear indicators that some of our models had issues with fitting to large datasets. For example, NSA, PSA, LR, and Random Forest models performed worse on larger datasets on almost every single metric. This is likely due to issues choosing relevant features from the larger datasets. For instance, NSA and PSA uniformly sample the larger subspace for detectors, which would become less suitable as the subspace to search grows from $\mathbb{R}^5$ to $\mathbb{R}^{45}$. Overall, XGBoost was the most performant method at scaling to large dataset sizes.

Regarding computational cost, nearly all models met the computational constraints set. Models which included control flows with function calls, such as NSA, PSA, and RF, had significantly higher latencies than models which could perform inference in a single function. RF models also had significantly larger RAM usage than other models, though this may be possible to mitigate by using quantised floating point values instead of the 64-bit values used in our models. Overall, LR and SVM models had similarly low inference latencies, which make them ideal for real-time applications. XGBoost also offers suitably-low latency, with the aforementioned improvements in performance.

Another benefit of note when working with constrained compute resources is that NSA, PSA, LR, and SVMs scale predictably and linearly in size as more features or detectors are added, making it simple to empirically determine the number of detectors to use for a particular system. Whereas RF and XGBoost models also have a similar property in that adding more estimators and depth grows the model size, it is harder to determine a precise size for a model–the number of nodes in each tree is capped by hyperparameters, but only set to some exact value by training.

Overall, we recommend future research explore bio-inspired AI algorithms other than NSA and PSA. For instance, an artificial immune system would be one candidate. An important note for research would be to modify the traditional search algorithm for detectors to be more performant in larger search spaces. As seen, XGBoost, LR, and SVMs are important benchmarks to assess the cost-benefit of the increased complexity of bio-inspired algorithms.

VII. Appendix

Supplementary materials are available online at https://github.com/Madhav-Malhotra/NSA-PSA-CUCAI

References

[1] H. Guo and J. Heidemann, "Detecting iot devices in the internet," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2323–2336, 2020.

[2] I. Ahmed, M. Anisetti, A. Ahmad, and G. Jeon, "A multilayer deep learning approach for malware classification in 5g-enabled iiot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1495–1503, 2023.

[3] J. Sengupta, S. Ruj, and S. Das Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, vol. 149, p. 102481, 2020.

[4] K. Collier, "50,000 security disasters waiting to happen: The problem of america's water supplies," *NBC News*, Jun 2021.

[5] D. E. Sanger, C. Krauss, and N. Pearlroth *The New York Times*, May 2021.

[6] S. Zeadally, E. Adi, Z. Baig, and I. A. Khan, "Harnessing artificial intelligence capabilities to improve cybersecurity," *IEEE Access*, vol. 8, pp. 23817–23837, 2020.

[7] S. Widup, A. Pinto, D. Hylender, G. Bassett, and P. Langlois, "2022 data breach investigations report," tech. rep., Verizon Communications Inc., 2022.

[8] D. Floreano and C. Mattiussi, *Bio-inspired Artificial Intelligence: Theories, methods, and technologies*. MIT PRESS, 2023.

[9] U. Rauf, "A taxonomy of bio-inspired cyber security approaches: Existing techniques and future directions," *Arabian Journal for Science and Engineering*, vol. 43, p. 6693–6708, Feb 2018.

[10] S. A. Hofmeyr and S. Forrest, "Architecture for an Artificial Immune System," *Evolutionary Computation*, vol. 8, pp. 443–473, 12 2000.

[11] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 202–212, 1994.

[12] D. Dasgupta and F. Nino, "A comparison of negative and positive selection algorithms in novel pattern detection," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0*, vol. 1, pp. 125–130 vol.1, 2000.

[13] H. Polat, M. Türkoğlu, O. Polat, and A. Şengür, "A novel approach for accurate detection of the ddos attacks in sdn-based scada systems based on deep recurrent neural networks," *Expert Systems with Applications*, vol. 197, p. 116748, 2022.

[14] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Information Fusion*, vol. 49, pp. 205–215, 2019.

[15] G. R. M. R., C. M. Ahmed, and A. Mathur, "Machine learning for intrusion detection in industrial control systems: Challenges and lessons from experimental evaluation," *Cybersecurity*, vol. 4, Aug 2021.

[16] P. Ioulianou, V. Vasilakis, I. Moscholios, and M. Logothetis, "A signature-based intrusion detection system for the internet of things," in *Information and Communication Technology Form*, (AUT), June 2018.

[17] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*, (Boston, MA), pp. 3971–3988, USENIX Association, Aug 2022.

[18] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home iot using openflow," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pp. 147–156, 2016.

[19] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for iot applications," *Wireless Personal Communications*, vol. 111, p. 2287–2310, Nov 2019.

[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009.

[21] I. Sharafaldin., A. Habibi Lashkari., and A. A. Ghorbani., "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP*, pp. 108–116, INSTICC, SciTePress, 2018.

[22] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," 2017.

[23] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search," 2012.