Gradient Boost enhanced Artificial Immune System algorithm for adaptive DDoS attack detection in IoT

Sayed Abualia
Department of Computer Sciences
Washington State University Vancouver
Washington, USA
sayed.abualia@wsu.edu

Anna Wisniewska
Department of Computer Sciences
Washington State University Vancouver
Washington, USA
anna.wisniewska@wsu.edu

Nirnimesh Ghose
School of Computing
University of Nebraska-Lincoln
Nebraska, USA
nghose@unl.edu

***Abstract*—Distributed Denial of Service attacks (DDoS) targeting the Internet of Things (IoT) remain a pervasive cybersecurity challenge. Biologically inspired solutions have shown promise for DDoS attack detection. For example, the human immune system has inspired various Artificial Immune System (AIS) solutions for anomaly detection. In this paper, we address the challenges of DDoS detection in IoT by proposing a Gradient boost regression and Adam-optimized Negative Selection Algorithm (GANSA). We show that the proposed algorithm is effective and can adapt to changes in network traffic patterns, thereby accurately detecting known and unknown DDoS attacks. We evaluate the proposed system against state-of-the-art machine learning DDoS detection algorithms (e.g., CNN, SVM). We show that the proposed system achieves a low false positive rate** (0.0003) **and near-perfect detection accuracy** (0.99)**, F1 score** (0.99)**, and MCC** (0.97) **while adapting to incoming network traffic in real-time.**

I. Introduction

The rapid growth of the Internet of Things (IoT) devices has facilitated many applications to improve quality of life [1]. As an increasing number of devices connect to the Internet, IoT networks become vulnerable to Distributed Denial of Service (DDoS) attacks [2]. Traditional security measures cannot cope with the sophistication and scale of these attacks [3]. Conventional intrusion detection methods generally do not dynamically adjust to new or previously unseen attack scenarios, limiting their effectiveness in a rapidly changing environment such as the Internet of Things (IoT), where the attack surface continuously expands. Moreover, anomaly detection algorithms often suffer from a high false positive detection rate where benign traffic is classified as malicious. Recently, machine learning algorithms have been used to detect sophisticated DDoS threats, since ML algorithms have the ability to analyze large volumes of data [4].

Deep learning–based detection algorithms have been shown to achieve high detection accuracy for diverse categories of anomalous traffic with varying sample sizes. For example, using Convolutional Neural Networks (CNNs) to analyze Internet of Things traffic has resulted in high detection accuracy of DDoS attacks [5]. In [6], an unsupervised learning method combined an expanded recurrent neural network (RNN) model with network flow properties to detect DDoS attacks. In [7], a deep learning approach was introduced that integrates softmax regression with a dense self-encoder. The approach achieved an average F-score of 75.76% in a 5-class detection framework when evaluated on the NSL-KDD benchmarking dataset.

A particularly promising subset of ML algorithms are bio-inspired algorithms modeled after natural processes and biological systems. Among bio-inspired methods, artificial neural networks, swarm intelligence, and artificial immune systems provide promising solutions for the cybersecurity domain [8]. These algorithms use optimization strategies observed in nature to improve the efficiency of cyber threat detection and handling. In [9], the authors showed how an artificial immune system (AIS) can learn from past attacks and adjust its security measures to detect emerging threats effectively. Unlike traditional methods that rely on a static threshold parameter for anomaly detection, AIS employs continuous learning from observed anomalies. The adaptive capability allowed the model to update dynamically, resulting in a reduced false alarm detection rate while increasing the likelihood of identifying novel anomalies in real-time. The effectiveness of machine learning algorithms based on swarm intelligence inspired by the cooperative behaviors of ants, bees, and other insects is discussed in [10]. These algorithms provide reliable protection against DDoS attacks through distributed and self-healing mechanisms. By learning from real-time data and dynamically adjusting parameters, these systems outperform traditional methods when processing volumetric and complex attacks, minimizing false positives, and enabling continuous adaptation to evolving attack patterns.

The Negative Selection Algorithm (NSA) and the Positive Selection Algorithm (PSA) are inspired by the immune system, where detectors are developed that match known signatures of abnormalities. NSA and PSA have been enhanced, for example, with advanced feature selection and adaptive threshold mechanisms to reduce false positive detection rates and improve the system's response to novel threats [11], [12]. However, some of the challenges with NSA-based methods remain, for example, its relatively complex computational requirements [11], and its high false positive detection rates caused by the presumption that any novel connection represents an intrusion [13].

In this paper, we propose a novel approach to enhance the efficiency and adaptability of the Negative Selection Algorithm (NSA). To mitigate NSA's shortcomings, we use Gradient

boost regression and Adam to optimize the Negative Selection Algorithm (GANSA). Adam optimization is utilized for initial parameter tuning in the training phase of the proposed model pipeline. Gradient Boost Regression (GBR) allows the negative selection algorithm to adapt to dynamic network traffic in the testing phase of the proposed model pipeline. The GBR and Adam optimization of the NSA results in real-time near-perfect detection accuracy of IoT DDoS attacks. Furthermore, the GANSA system can adapt to novel traffic and detect zero-day DDoS attacks.

II. System Model and Problem Definition

The architecture of the DDoS detection system is illustrated in Fig.1. The proposed intrusion detection system (IDS) is implemented on a server (following [5]). The network gateway receives traffic from the outside network and relays it to the server, which collects traffic features such as port number, IP address, network protocol, connection frequency, and transmission flow. The server classifies the traffic as benign or malignant in real-time. The gateway relies on this classification to forward only benign traffic to the internal network.

The overall speed of the internal network is boosted as the bottleneck created by overburdened IoT devices is alleviated. Hence, delays due to useless traffic processing are avoided as data moves across the network, where only relevant traffic reaches the IoT devices, better utilizing the available network bandwidth and device resources. In this way, the network environment is maintained in a stable and reliable state.

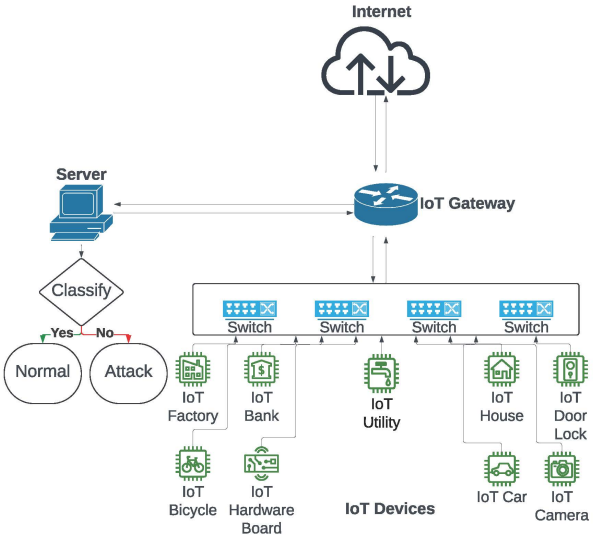

Fig. 1. IoT System Model

A. Threat Model

Distributed Denial of Service (DDoS) attacks are malicious attempts to disrupt the normal functioning of a targeted system, network, or service by overwhelming it with a flood of incoming traffic. These attacks exploit vulnerabilities in the target's infrastructure, exploiting weaknesses in network protocols, servers, or applications to exhaust resources and render the service inaccessible to legitimate users.

Any outside entity can launch a DDoS attack on the network, primarily utilizing a botnet. Infected computers, servers, or IoT devices, when joined together, create botnets that an attacker can control from a distance to bombard the target with traffic. DDoS attacks appear in many forms, all of which attack different layers of the network stack or capitalize on particular vulnerabilities. In volumetric attacks consisting of UDP and ICMP floods, the target experiences a saturation of traffic, which results in network congestion. Network protocol weaknesses allow protocol attacks such as SYN floods and Ping of Death to consume resources. HTTP floods and DNS amplification are attacks on the application layer that either consume server resources or disrupt specialized services.

B. Problem Definition

Detecting and combating DDoS attacks proves challenging because of their distributed and dynamic characteristics. Traditional mitigation techniques comprise of traffic filtering with intrusion detection/prevention systems or firewalls, rate limiting to mitigate volumetric assaults, and employing machine learning or statistical approaches to recognize irregularities in traffic patterns. Content Delivery Networks (CDNs) work to mitigate DDoS attacks by pushing content across numerous servers, dealing with and diminishing the attacks near their origin.

We explore the following research questions:

RQ:1 How to efficiently perform DDoS attack detection in IoT networks as the network traffic changes dynamically?

RQ:2 How to reduce the false alarm rate, allowing benign traffic through the gateway without affecting the detection rate?

To answer the questions above, we show that using Gradient boost regression and Adam to optimize the Negative Selection Algorithm (GANSA) allows for dynamic detection of DDoS attacks with high detection accuracy while keeping the false alarm rate low. Adam optimization is used for initial parameter tuning in the training phase of the proposed model pipeline. Gradient Boost Regression (GBR) is used to allow the negative selection algorithm to adapt to dynamic network traffic in the testing phase of the proposed model. The GBR and Adam optimization of NSA results in near-perfect detection accuracy of IoT DDoS attacks in real-time.

III. Proposed System

The proposed system pipeline, shown in Figure 2, consists of the following three steps: (1) Dataset preprocessing, (2) Training phase using Adam together with the Negative Selection Algorithm (NSA), (3) Testing phase using Gradient Boost Regression (GBR) together with the Negative Selection Algorithm (NSA). In what follows, we describe each of the phases of the pipeline.

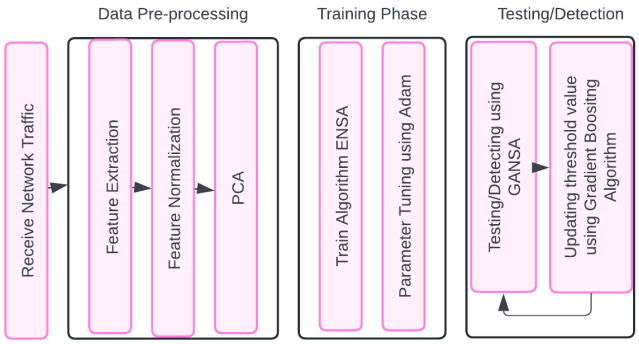

Fig. 2. Framework of the proposed GANSA pipeline.

A. Data pre-processing

To effectively evaluate the proposed GANSA model, we used the NSL-KDD dataset. The NSL-KDD dataset is a benchmark dataset for network intrusion detection. It consists of manually pre-labeled data for supervised learning, framing attacks under different types, and qualitatively ranking them across 41 features. The structured attributes allow for extensive analysis and testing of various machine learning algorithms in the cybersecurity domain.

To facilitate comparison with the CNN-based IDS in [5], we used the same methodology to preprocess the dataset. In the NSL-KDD training dataset, each record is categorized as benign or DDoS attack using 41-dimensional features, including 3-dimensional non-characteristics features and 38-dimensional characteristics features. The three steps for data preprocessing are described next. First, one-hot encoding is utilized to represent the four non-characteristic features (protocol type, service, flags, and class) as binary vectors. The result of each record in the dataset consists of 122-dimensional character features. Second, the input values are normalized

$$y = \frac{y - M_{min}}{M_{max} - M_{min}} \tag{1}$$

where y is the value to be normalized, M_{min} is the minimum number in a dimension, and M_{max} denotes the maximum number in a dimension. Finally, to reduce similar features and avoid data over-fitting, Principal Component Analysis (PCA) is used to train the dataset. This results in the reduction of features in each record from 122 to 58 and in an improvement of training time for the proposed model.

B. Training phase: NSA-Adam

The biological immune system has inspired Artificial Immune System (AIS) computational models for anomaly detection. One of the main mechanisms of the immune system is the discrimination between the non-self and the self. The Negative Selection Algorithm is a computational imitation of this process. The inspiration for the NSA algorithm comes from the immune system's T-cell maturing process [14]. The essential idea is to randomly generate a diverse set of T-cells where those T-cells that recognize self-cells are eliminated while the rest (detectors) are deployed into to immune system to recognize outside pathogens (nonself). In the context of networks, a detector set is used to identify the incoming network traffic as self or nonself. If the incoming traffic matches the detector, it is considered nonself, i.e., an anomaly (DDoS attack).

In NSA, matching rules (where the distance measure between the detector and the data instance is within a threshold) are used in the anomaly detection phase and the detector generation phase. The matching rule M is defined as follows:

$$\mathrm{M}(x) = \begin{cases} \text{Attack} & \text{if } \exists d \in D \text{ similarity}(x, d) \geq \epsilon \\ \text{Benign} & \text{otherwise} \end{cases} \tag{2}$$

where D is the set of detectors, x is the data instance and ϵ is the matching threshold. For the $similarity(\cdot)$ function we use Euclidean distance

$$\text{similarity}(x, d) = \sqrt{\sum_{i=1}^{m} (d_i - x_i)^2} \tag{3}$$

where m is the dimensionality of the feature space.

In the training phase, we use NSA together with the adaptive moment estimation (Adam) algorithm to optimize the GANSA intrusion detection system. We use Adam optimizer for NSA parameter tuning due to its computational efficiency and its ability to handle large datasets, making it particularly suitable for dynamic network environments. Through iterative updates, Adam calibrates the NSA parameters to ensure that the detection error is minimal [15]. For each iteration t, Adam computes the gradients of the loss function $J(\theta)$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla J(\theta_t) \tag{4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla J(\theta_t))^2 \tag{5}$$

where θ is the weights and biases, α is the learning rate, β_1 is the exponential decay rate for the first moment estimate, β_2 is the exponential decay rate for the second moment estimate, m is the first moments of the gradients, and v is the second moment of the gradients. Adam corrects the bias in m and v estimates as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{6}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{7}$$

Finally, the parameters are updated using the corrected estimates:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \kappa} \tag{8}$$

where κ is a constant. Adam is capable of handling dense and difficult gradients. The training cycle is performed repeatedly until the NSA parameters are optimized. Incorporating Adam optimizer in the training phase to tune the NSA parameter significantly improves the performance of the original NSA.

C. Testing phase: NSA-Gradient Boost Regression

The challenge with the NSA is that it does not adapt efficiently to dynamic network traffic due to the static nature of the threshold parameter ϵ. To address this shortcoming, we use Gradient Boost Regression (GBR) to periodically tune the threshold parameter ϵ. Gradient boosting regression is a powerful ensemble learning technique widely used in machine learning for both classification and regression tasks [16]. It is known for its high predictive accuracy, robustness to overfitting, and flexibility in handling heterogeneous data.

In the testing phase, the NSL-KDD testing dataset is divided into equally sized partitions P. We adjust the threshold parameter ϵ in the NSA by running the GBR algorithm after each partition $p \in P$. For each partition p, vector S is obtained consisting of the following measures across features: mean (μ), standard deviation (σ), minimum($\min$), maximum ($\max$), median (Med), skewness (Skew), kurtosis (Kurt), 25th percentile (Q_1), 75th percentile (Q_3), interquartile range (IQR = $Q_3 - Q_1$), and coefficient of variation (CV = $\frac{\sigma}{\mu}$). We trained the GBR algorithm by using the aggregated distance e of two successive vectors S_{i-1} and S_i. We obtain the absolute differences of each statistic in S for successive partitions p_i and p_{i-1}

$$S_i = |S_i - S_{i-1}| \tag{9}$$

where $i > 1$. The aggregated distance for vector S_i is defined as follows

$$e_i = \sum_{j=1}^{k} S_{ij} \tag{10}$$

where k is the number of elements in vector S. We update the threshold parameter ϵ in the NSA after each partition p_i as follows:

$$\epsilon_i = \epsilon_{i-1} \cdot (1 + A_i) \tag{11}$$

where A_i is the predicted adjustment factor obtained through GBR after partition p_i using e_i. Algorithm 1 shows the NSA with gradient boost regression threshold optimization.

Algorithm 1 NSA with GBR Threshold Adaptation

Initialize $\epsilon_i \leftarrow \epsilon_0$ (where ϵ_0 is obtained in the GANSA training phase), $S_i \leftarrow \emptyset$, $i \leftarrow 1$ (where i is the current partition number), $j \leftarrow i$
while True **do**
  Run NSA(ϵ_j) on incoming network traffic
  if $S_i \neq \emptyset$ **then** Partition i finished running
    $S_i \leftarrow |S_i - S_{i-1}|$
    $e_i \leftarrow \sum_{j=1}^{k} S_{ij}$
    $A_i \leftarrow GBR(e_i)$ Run GBR with e_i
    $\epsilon_i \leftarrow \epsilon_{i-1} \cdot (1 + A_i)$ Update the threshold
    $j \leftarrow i$
    $i \leftarrow i + 1$
  end if
end while

IV. Results

Our experiments were carried out using 4 Nodes, an Intel Xeon Gold 6230 with a maximum frequency of 2.10 GHz, 128GB of RAM, and one GPU Nvidia H100 running on Rocky Linux 9.1 (Blue Onyx). To evaluate the GANSA model, we used the NSL-KDD dataset [17]. The NSL-KDD dataset is a benchmark for intrusion detection. NSL-KDD is divided into training and testing datasets, KDDTrain+ and KDDTest+ respectively. The KDDTest+ dataset has overlapping records with the KDDTrain+ dataset as well as unique records to facilitate the evaluation of zero-day attacks. Following [5], we extracted 113,270 records related to IoT traffic from the KDD Train+ dataset including DDoS attacks such as spy, rootkit, buffer_overflow, nmap, land, and back. We extracted 22,545 records from the KDD Test+ dataset including DDoS attacks such as phf, processtable, snmpgetattack, ftp_write, saint, apache2, and buffer_overflow. The testing dataset was divided into 10 partitions P where each partition was of size 2.56 MB.

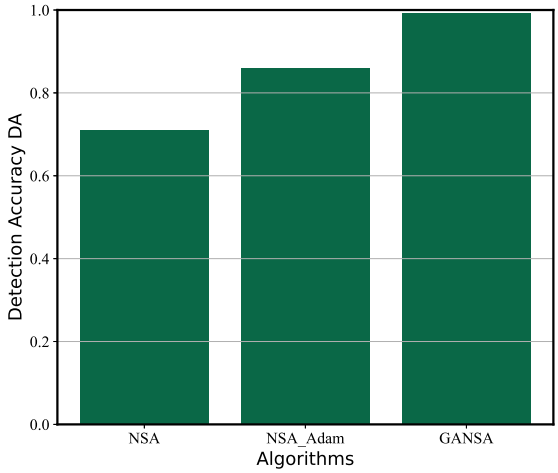

Fig. 3. Detection accuracy DA for NSA, NSA Adam, and GANSA.

We begin by evaluating the effect that Adam and Gradient Boost Regressions have on NSA in the GANSA testing/training pipeline. The confusion matrix describes the outcome of the classification task where

- True Positive (TP) denotes DDoS attacks that are correctly classified as DDoS attacks.
- False Negative (FN) denotes DDoS attacks that are incorrectly classified as benign traffic.
- True Negative (TN) denotes benign traffic correctly classified as benign traffic.
- False Positive (FP) denotes benign traffic incorrectly classified as DDoS attacks.

The detection accuracy DA is defined as follows:

$$DA = \frac{TP + TN}{TP + TN + FP + FN}. \tag{12}$$

In Figure 3, we compare the detection accuracy of NSA, NSA with Adam, and GANSA (NSA with GBR during testing and Adam during training). We can see that Adam improves the performance of NSA where the original NSA algorithm has a detection accuracy of $DA = 0.71$ while NSA with Adam has a detection accuracy of $DA = 0.86$. In addition, we can see that the Gradient Boost Regression algorithm further improves the detection accuracy where GANSA has a detection accuracy of $DA = 0.99$.

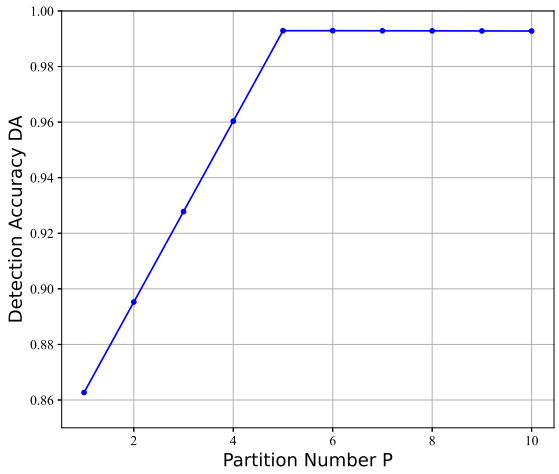

Fig. 4. Detection Accuracy DA vs. Partition Number P

To evaluate the adaptive nature of GANSA, we observe the detection accuracy DA after each partition p. In Figure 4, the x-axis shows the partition number P, while the y-axis shows the detection accuracy DA for the GANSA intrusion detection system. We can see the detection accuracy steadily increases up to the 5th partition, indicating that the algorithm is adjusting to incoming network traffic. After the 5th partition, the detection accuracy converges and reaches a near-ideal accuracy of $DA = 0.99$. In other words, the GANSA system has a fast convergence time of 5 partitions and remains stable with respect to detection accuracy as new traffic is received. When comparing GANSA to other ML algorithms, we take into account both conv-GANSA (GANSA including the convergence period) and GANSA post-convergence.

We compare the GANSA IDS to the CNN-based approach in [5] referred to as CNN-SSL as well as the following state-of-the DDoS detection machine learning algorithms: SVM, DT, BAYES, KNN, and RNN. In addition to the detection accuracy, we use false alarm rate, precision, recall, specificity, and F1 score to evaluate GANSA, defined as follows:

$$FalseAlarmRate(FAR) = \frac{FP}{FP + TN} \tag{13}$$

where FAR is the fraction of benign traffic incorrectly classified as DDoS attacks overall benign traffic.

$$Pricision = \frac{TP}{TP + FP} \tag{14}$$

where precision is the fraction of correctly identified DDoS attacks out of all traffic classified as DDoS attacks.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

where recall, sometimes referred to as sensitivity, is the fraction of DDoS attacks that are detected out of all DDoS attempts.

$$Specificity = \frac{TN}{TN + FP} \tag{16}$$

where Specificity is the fraction of correctly identified benign traffic out of all benign traffic.

$$F1Score = \frac{2 * Pricision * Recall}{Pricision + Recall}. \tag{17}$$

where the F1 score is the weighted average of recall and precision.

In Table I, we can see the performance metrics for GANSA, and conv-GANSA compared to other ML DDoS detection algorithms. We can see that GANSA has a near-perfect accuracy of 0.99 for detection, precision, recall, specificity, and the F1 score. Furthermore, GANSA has the lowest false detection rate $FAR = 0.0003$ when compared to the other detection algorithms. Even when observing conv-GANSA (i.e., including the convergence stage), it outperforms the other ML algorithms when comparing detection accuracy DA and F1 score. KNN has lower FAR rate (0.005) and higher specificity (0.99) than conv-GANSA, while CNN has higher recall (0.97) than conv-GANSA. Although, for the overall performance, conv-GANSA outperforms the other ML approaches.

TABLE I
PERFORMANCE EVALUATION.

	CNN	SVM	DT	BAYES	KNN	RNN	GANSA	conv-GANSA
Precision	0.87	0.84	0.72	0.98	0.98	0.98	0.99	0.98
Recall	0.97	0.94	0.96	0.71	0.73	0.80	0.99	0.95
F1 score	0.92	0.88	0.83	0.82	0.84	0.88	0.99	0.96
Accuracy	0.92	0.85	0.81	0.86	0.88	0.87	0.99	0.96
Specificity	0.87	0.83	0.68	0.98	0.99	0.98	0.99	0.97
FAR	0.120	0.168	0.318	0.016	0.005	0.016	0.0003	0.022
AUC	0.90	0.82	0.70	0.85	0.86	0.89	0.99	0.96
MCC	0.85	0.78	0.66	0.73	0.77	0.80	0.97	0.93

We further evaluate the performance of the GANSA IDS by looking at the Receiver Operating Characteristic (ROC-AUC) score and the Matthews Correlation Coefficient (MCC). The ROC curve is a plot between True Positive Rate (TPR) and False Positive Rate (FPR) at varying threshold values. In Figure 5, we can see that the GANSA system outperforms the other ML detection algorithms. In Table I, we can see that the proposed GANSA approach has $AUC = 0.99$. The MCC score captures the performance of the algorithm when detection accuracy and F1 score are not adequate measures of the performance due to an imbalanced dataset. MCC is defined as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN).}} \tag{18}$$

In Table I, we can see that even here GANSA outperforms the other ML algorithms with a score of $MCC = 0.97$.

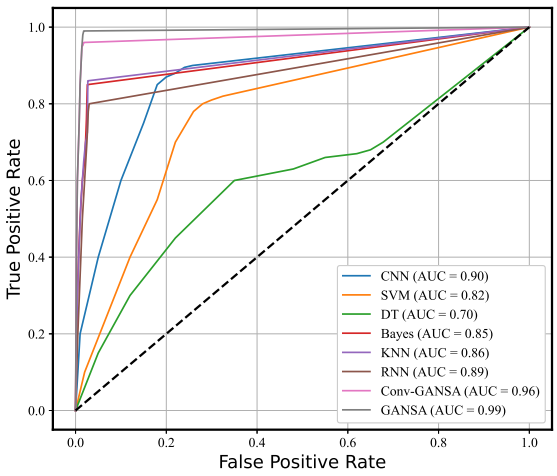

Fig. 5. ROC-AUC Curve

Finally, we observe the run time for GANSA. Since the GBR algorithm is invoked after every partition p, there is a trade-off between the number of partitions, detection accuracy, and the testing run time. Through experimentation, we found that when $P = 10$ we achieve the perfect balance between detection accuracy and run time. In Table II, we can see that the testing phase run time (0.698 seconds) for GANSA is significantly lower when compared to the other ML detection algorithms (e.g., CNN with run time of 1.24 sec.). The rapid

TABLE II
RUN TIME

Algorithm	**Run Time/s**	
	Training	***Testing***
CNN	42.17	1.24
SVM	88.54	4.87
KNN	37.12	5.25
RNN	23.99	1.46
Classic NSA	23.44	12.38
GANSA	19.93	0.698

testing run time for GANSA indicate that the proposed system can detect DDoS attacks in real-time.

V. CONCLUSION

To combat Distributed Denial of Service attacks in the Internet of Things, we have proposed a novel Gradient boost regression and Adam optimized Negative Selection Algorithm (GANSA). The proposed GANSA system is both an effective and adaptable solution for DDoS detection in IoT. To evaluate our proposed model, we used the NSL-KDD dataset and compared it to numerous state-of-the-art machine-learning DDoS detection algorithms including CNN, SVM, DT, and NSA. We showed that the proposed GANSA system outperformed the other ML algorithms with outstanding detection accuracy (0.99), precision (0.99), recall (0.99), F1 score (0.99), and specificity (0.99) while maintaining low false alarm rate (0.0003). Furthermore, we obtained the ROC-AUC curve and MCC score (to account for e.g., an unbalanced dataset) where GANSA achieved the highest scores of (0.99) and (0.97) respectively when compared to the other ML DDoS detection algorithms. Finally, we showed that given the adaptive nature of the GANSA algorithm (by utilizing gradient boost regression to adjust the NSA threshold parameter ϵ), we can achieve the detection of novel DDoS attacks in real-time.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. The UNL portion of the work was supported by NSF Grants CNS-2225161. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

REFERENCES

[1] S. Nižetić, P. Šolić, D. L.-d.-I. Gonzalez-De, L. Patrono *et al.*, "Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future," *Journal of cleaner production*, vol. 274, p. 122877, 2020.

[2] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in iot: a survey," *The Journal of Supercomputing*, vol. 76, pp. 5320–5363, 2020.

[3] R. Vishwakarma and A. K. Jain, "A survey of ddos attacking techniques and defence mechanisms in the iot network," *Telecommunication systems*, vol. 73, no. 1, pp. 3–25, 2020.

[4] A. Nassar and M. Kamal, "Machine learning and big data analytics for cybersecurity threat detection: A holistic review of techniques and case studies," *Journal of Artificial Intelligence and Machine Learning in Management*, vol. 5, no. 1, pp. 51–63, 2021.

[5] L. Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A deep learning-based ddos detection framework for internet of things," in *Proc. of IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[6] C. G. Cordero, S. Hauke, M. Mühlhäuser, and M. Fischer, "Analyzing flow-based anomaly intrusion detection using replicator neural networks," in *Proc. of Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016, pp. 317–324.

[7] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 21–26.

[8] F. Saruhan-Ozdag, D. Yiltas-Kaplan, and T. Ensari, "Detection of network attacks with artificial immune system," in *Pattern Recognition Applications in Engineering*. IGI Global, 2020, pp. 41–58.

[9] M. Dimolianis, A. Pavlidis, and V. Maglaris, "Signature-based traffic classification and mitigation for ddos attacks using programmable network data planes," *IEEE Access*, vol. 9, pp. 113 061–113 076, 2021.

[10] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453–563, 2022.

[11] P. Saurabh and B. Verma, "Negative selection in anomaly detection—a survey," *Computer Science Review*, vol. 48, p. 100557, 2023.

[12] H. Alrubayyi, G. Goteng, M. Jaber, and J. Kelly, "A novel negative and positive selection algorithm to detect unknown malware in the iot," in *Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2021, pp. 1–6.

[13] M. E. Pamukov, V. K. Poulkov, and V. A. Shterev, "Negative selection and neural network based algorithm for intrusion detection in iot," in *Proc. of International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2018, pp. 1–5.

[14] K. D. Gupta and D. Dasgupta, "Negative selection algorithm research and applications in the last decade: A review," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 110–128, 2021.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proc. IEEE symposium on computational intelligence for security and defense applications*. IEEE, 2009, pp. 1–6.