Intrusion Detection Based on Generative Adversarial Network of Reinforcement Learning Strategy for Wireless Sensor Networks

Jun Tu
School of Computer, Hubei University of Technology, Wuhan 430068, Hubei, China

Willies Ogola
School of Computer, Hubei University of Technology, Wuhan 430068, Hubei, China

Dehong Xu
Wuchang Shipbuilding Industry Group Co. Ltd., Wuhan 430068, Hubei, China

Wei Xie
School of Computer, Hubei University of Technology, Wuhan 430068, Hubei, China

Received: March 9, 2021. Revised: October 21, 2021. Accepted: November 18, 2021.

***Abstract*—Due to the wireless nature of wireless sensor networks (WSN), the network can be deployed in most of the unattended environment, which makes the networks more vulnerable for attackers who may listen to the traffic and inject their own nodes in the sensor network. In our work, we research on a novel machine learning algorithm on intrusion detection based on reinforcement learning (RL) strategy using generative adversarial network (GAN) for WSN which can automatically detect intrusion or malicious attacks into the network. We combine Actor-Critic Algorithm in RL with GAN in a simulated WSN. The GAN is employed as part of RL environment to generate fake data with possible attacks, which is similar to the real data generated by the sensor networks. Its main aim is to confuse the adversarial network into differentiating between the real and fake data with possible attacks. The results that is from the experiments are based on environment of GAN and Network Simulator 3 (NS3) illustrate that Actor-Critic&GAN algorithm enhances security of the simulated WSN by protecting the networks data against adversaries and improves on the accuracy of the detection.**

***Keywords*—Intrusion Detection, Machine Learning Algorithm, Reinforcement Learning, Wireless Sensor Networks, Generative Adversarial Networks.**

I. INTRODUCTION

Wireless sensor networks were originally conceived for military applications like battlefield monitoring. But today these networks have many purposes in both industry and consumer applications. Applications like environmental monitoring, smart homes, industrial process control and machine health monitoring are just a few examples of areas where wireless sensor networks are being put to use. The growth of applications and the constant evolution of cheaper and more capable hardware make wireless sensor networks an interesting domain with a lot of potential research opportunities [1]. A wireless sensor network is typically composed of multiple autonomous wireless sensor nodes. These nodes gather data about their environment and work together to forward their data to centralized locations called base stations or sinks. The sensor nodes are equipped with various sensors that allow them to monitor their environment. The type and amount of sensors that are available on the sensor nodes depends on the application. Some of the common examples are sensors for measuring temperature, humidity, movement, visible light, and so forth. The data that each node gathers from its sensors is passed along through the wireless network to special nodes that are responsible for collecting data. These nodes usually do not perform any measurements themselves and are called base stations or sink nodes. The sink nodes forward the collected data to a different system that they are connected to. This can be a directly connected computer or a device on a second network interface that the sink could be connected to [2]. Sensor nodes are deployed densely in a Wireless Sensor Network to increase the accuracy of

collected results as the target area may be too large to collect accurate readings with one or two sensor nodes. Multiple sensor nodes in most cases are able to detect a single target of interest at the same time thus leading to a highly correlation and redundancy in collected data. If each node in the sensing region sends data to the sink, a lot of energy will be depleted fast. There is therefore the need to minimize the redundant data as we do not need similar readings to be transmitted to the sink node; we only need an accurate reading being transmitted from a particular region.

II. RELATED WORK

A. *Literature* on *Generative Adversarial Networks use* in *intrusion detection tasks*

In a recent literature, the author introduced a spoofing attack by an adversary pair of a transmitter and a receiver that assume the roles of the generator and discriminator in the GAN and using the game theory, plays a mini-max game by generating the best spoofing signals which aim to fool the best trained defense mechanism. From the point of view of an attacker, the deep learning-based spoofing mechanism is trained to potentially fool a defense mechanism, in this case, a Radio Frequency fingerprinting system. From the view point of the defender, the deep learning based defense mechanism is trained against potential spoofing attacks when an adversary pair of a transmitter and a receiver co-operates. GAN optimization goal is to reach Nash equilibrium. This basically means that the optimization terminates at a saddle point, that is, achieves a minimum with respect to the generator and a maximum with respect to the discriminator [3]. GAN have been shown to successfully approximate high dimensional data,the following formula 1 shows an object of training between the discriminator(D) and the generator(G).

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[logD(x)] + \mathbb{E}_{x\sim P_z(z)}[log(1-D(G(z)))] \quad (1)$$

GAN based spoofing attack showed an increase in the probability of success of the wireless signal [4]. A large number of works utilize Bidirectional GANs (BiGANs) [5] to detect anomalies/intrusions. The authors trained an encoder network to transform data samoles to latent variables at the moment of testing. The anomalies are detected through the discrepancy between the testing sample and the regenerated sample [5]. Literature [6] combining BiGAN and anomaly detection of combining site, improve traditional generator generated against network single rely on determination of feedback for the shortage of the gradient update, to distinguish from the traditional discriminant generation gap between sample and real sample as normal loss will be converted to real data and real data is mapped to a hidden under the space sampling to generate samples again after the gap between Determine whether the test sample is abnormal. Literature [7] dispalys a new type of imbalanced generative adversarial network (IGAN) is proposed to solve the class imbalance problem. The main innovation of the model is to introduce a new unbalanced data filter and convolution layer into the generative adversarial network structure to generate new samples for a few classes.

B. *Literature on Reinforcement learning use intrusion detection tasks*

In recent years, strategies of reinforcement learning have applied in WSNs for numerous reasons. A multi-Agent RL based routing has been proposed which uses cooperative communication for enhanced quality of Service and to prolong the lifetime of the network.

An internet of thing (IOT) on intrusion detection mechanism based Harmony Search Hawks Optimization and deep Reinforcement Learning [8]. Among them, deep Reinforcement Learning classifier detects the malicious or intruder behaviors effectively.The literature [9] combine wireless software-defined-networking(W-SDN) with actor-critic algorithm in Reinforcement Learning.It constructs a state representation of observed environment status, a low dimension reponse matrix as well as a simplified response selection policy.Its simulation results reveal that benefiting from local environment observation and reach a goal in effective intrusion response without sophisticated feature engineering. The paper [10] reviews pre-existing surveys during the application of RL and deep reinforcement learning (DRL) techniques in IOT communication technologies.Its emphasis is on the analysis on applying RL and DRL techniques in wireless IOT to resolve issues related to routing,sheduling,etc.Network Intrusion Detection (NID) exists an import issue is the model design and prediction of real-time online data composed of a series of time-related feature patterns.Literature [11] empolyes an RL that includes a deep auto-encoder in the Q-network (DAEQ-N). DAEQ-N model attempts to reach the maximum prediction accuracy in online learning systems for classifying normal or anomaous behaviors.

C. *Connection Between Generative Adversarial Networks (GANS) and Reinforcement Learning (RL)*

Over the years, hybrid models have shown great potential in solving a diverse array of Machine Learning (ML) and Artificial Intelligence (AI) problems. Figure 2 displays Actor-Critic Algorithm in Reinforcement learnning. Actor-Critic Algorithm is made up of both actor network and critic network in the custom environment. While most algorithm in Reinforcement Learning focus on learning the Q-value such as in Q-learning and some learning as Policy function as in Policy Gradients. Actor-Critic methods try to learn both the Policy and the Q-value simultaneously.The critic estimates the value function while the actor estimates the policy.

Generative Adversarial Networks and the Actor-Critic (AC) method in Reinforcement Learning is an example of networks which have shown close parallels. They have shown really strong connection between their two classes of models [12]. The connection between AC and GAN is as follows.

Considering a RL MDP problem [12], its parameters are including the collections with states S and behaviors A, the initial state distribution for the $P_0(s)$,transfer function $P(s_{t+1}|s_t,a_t)$, the reward distributio $R(s_t)$,and the discount factor $\gamma \in [0,1]$. The purpose of AC algorithm is synchronously learn a behavior value function

$Q^{\pi}(s,a)$,and predict discount reward expections such as formula (2).The following formula (2), (3), (4) [12] is able to expressed.

$$Q^{\pi}(s,a) = \mathbb{E}_{s_{t+k}\sim P, r_{t+k,}\sim R, a_{t+k}\sim\pi}[\sum_{k=1}^{\infty}\gamma^{k} r_{t+k}\,|s_t = s, a_t = a] \quad (2)$$

Then,objectives learn an optimal strategy according to the value function such as formula (3).

$$\pi^{*} = arg\max_{\pi}\mathbb{E}_{s_0\sim p_0, a_0\sim\pi}\,[Q^{\pi}(s_0,a_0)] \quad (3)$$

Among $Q^{\pi}(s_0,a_0)$ is the solution of the minimum problem,which is equivalent to formula (4), where $\mathcal{D}(\cdot\,|\,\cdot)$ represents the divergence between arbitrary distributions.

$$Q^{\pi} = arg\min_{Q}\mathbb{E}_{s_t,a_t\sim\pi}[\mathcal{D}(\mathbb{E}_{s_{t+1},r_t,a_{t+1}}[r_t + \gamma Q(s_{t+1},a_{t+1})]||Q(s_t,a_t))] \quad (4)$$

The connection between GAN and Actor-Critic is considered precise. GAN can be interpreted as a updated AC algorithm using a Markov Decision Process (MDP). Each feature from Generated samples can be equivalent to each action. If environment choose real feature that is belong to real samples, the Agent of RL would get a reward. Training process of the original GAN is a game and difficult to measure accurately along with the time. The saddle point of GAN always be hard to reach and the training time of GAN is long.

In order to effectively ease the problems which include accuracy of intrusion detection using the united technologies between GAN and RL, the Section III is the proposed algorithm using exploring the effectiveness of intrusion detection in simulated environment of RL and GAN.

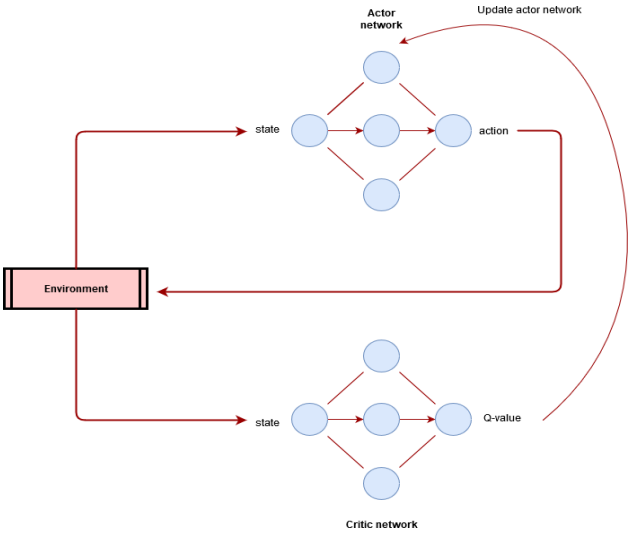

Figure 1:Architecture of Actor-critic.

III. THE SPECIFIC DETAILS

In our pipeline, RL is used to control the complex latent space of the GAN. The proposed algorithm trains an actor-critic based network by learning the policy in a continuous action space by adopting the Deep Deterministic Policy Gradient (DDPG) proposed by Lillicrap et al. [13] for the optimization of our actor critic method. In the DDPG framework, the actor network is parameterized in a deterministic manner, it learns a policy and maps the states to a particular action. [13] The critic network on the other hand, Q (s, a), uses the Bellmans equation to provide a measure of the quality of action and that of the state. We assume that our environment is modelled as a MDP meaning that the current state and actions depend on the previous state and action. Our environment can also be said to be fully observable. In our setting, the environment is our GAN network and the action. It takes can be defined as the input to the generator in the GAN network. One of the main key task in training an RL agent is choosing the correct formulation of the reward function. One of the key tasks in the training of an RL agent is formulating the correct reward function. The reward in RL at any given state is the discounted future reward using formula (5):

$$R_t = \sum_{i=t}^{T}\gamma^{(i-t)}\, r(s_i,a_i) \quad (5)$$

The main task is to find a policy which provides the maximum reward. Depending on the quality of the action the agent takes, the environment gives a reward (R) back to the RL agent. In our proposed algorithm, the quality of action equates to the correct seed selection input our RL algorithm feeds into the generator. Rewards will be awarded by the same RL model to the discriminator based on how accurately it is in calculating the probability of the real data from fake data. This means that the model ought to have access to the real distribution and the fake distribution generated by the GAN. Our GAN network ought to filter out fake data or detect anomalies in the data and only allow real data generated by the sensor nodes to pass through. Before training the RL agent, we make sure that the GAN is pretrained well as it constitutes as part of the RL environment. The detailed algorithm process can be viewed in Figure 2.

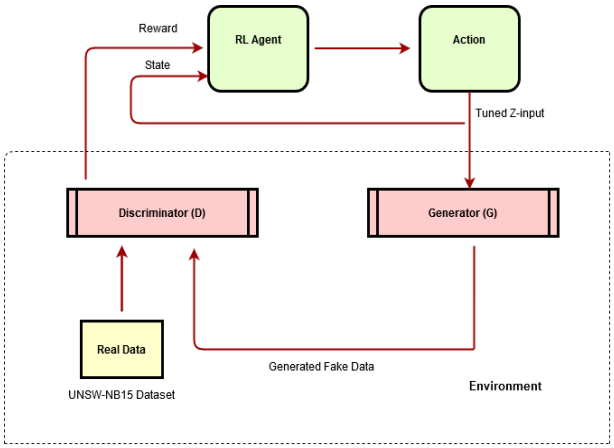

Figure 2: The proposed architecture of our algorithm.

IV. EXPERIMENT EVALUATION

A. *Experiment setting*

The new novel algorithms framework is implemented in Python and all the experiments are performed using the Keras [14] library. Keras is one of the best high level neural networks APIs for deep learning which sits on top of the

Tensorflow ML framework. It is written in Python programming language and supports multiple back-end neural network computational engines. Keras supports the scikit-learn library features such as cross validation and grid search. The algorithm was evaluated on the UNSW-NB15 [15] datasets containing a training set of 175,341 samples and a testing set of 82,332 samples. The table 1 and table 2 shows detailed information of UNSW-NB15 datasets.

Dataset	Total
Training set	175,341
Testing set	82,332

TABLE 1. numbers of UNSW-NB15 datasets

Attack Type	Training Set	Testing Set
Fuzzer	18,184	6,062
Analysis	2,000	677
Backdoor	1,746	583
Dos	12,264	4,089
Exploits	33,393	11,132
Generic	40,000	18,871
Shellcode	1,133	378
Worms	130	44
Reconnaissance	10,491	3,496

TABLE 2. the distribution of all records of the UNSW-NB15 dataset

All experiments were performed in the NS3 toolkit, a discrete simulation environment for Wireless Sensor Networks. This environment has been developed and tested in the C++ and Python programming software environment. In order to validate the simulator we tested multiple well known algorithms and verified that their behavior was in line with expections. The main goal of the simulation is to monitor the wireless sensor network and secure the communication of data from either internal or external attacks. The discriminator network needs to be robust enough to ensure the network is fully protected against attacks or malicious nodes.

All nodes have an identical fixed range in which they can communicate with neighbors. Because we use a two dimensional model we can calculate the communication ranges as the Euclidean distance between nodes. The system does not simulate noise and assumes a perfect link quality between all nodes that are within the communication range of each other. Communication between neighboring nodes takes one discrete time step. This means that a message can travel one hop per time step.

In our NS3 simulator, the size of the network is 1500 by 1500 m2 network topology. The network involves 150 sensor nodes with a transmission range of 40 meters. The initial energy of the nodes is set to 6 joules. The maximum energy consumption of the sensor nodes for receiving (Rx) and transmitting (Tx) the data is set to 14 mW and 13.0 mW, respectively. Sensing and idle nodes have 10.2mWand 0.42mW, respectively. The maximum simulation time is 45 minutes.

B. T-SNE Visualization

Developed by Laurens van der Maatens and Geoffrey Hinton in 2008, T-distributed Stochastic Neighbor Embedding(t-SNE) is an unsupervised, non-linear, machine learning algorithm for data exploration and visualization used on very large data. It is usually best suitable for embedding a high dimensional data for visualization in a low-dimensional space of two (2D) or three dimensions (3D) [16]. The t-SNE algorithm calculates a similarity measure between pairs of instances in the low dimensional space and in a high dimensional space. After which it tries to optimize these two similarity measures using a cost function. The ML algorithm, t-SNE has the capability of capturing much of the local structure of a high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales. The distance between both distributions, that is, normal and t-distributions, are minimized by using the kullback leibler (KL) divergence as its cost function. The t-SNE algorithm, therefore, attempts to find patterns in the data by identifying observed clusters based on the similarity of data points which have multiple features. The algorithm builds a probability distribution over pairs of high-dimensional data in a way such that data which shows a lot of similarity have a high probability of being selected, while data which appear dissimilar, have a rather small probability of being selected. The t-distribution has shown to be faster in the evaluation of data and has the ability to withstand outliers. Figure 2 above shows the t-SNE scatter plot of the UNSW-NB15 dataset projected onto two dimensions.

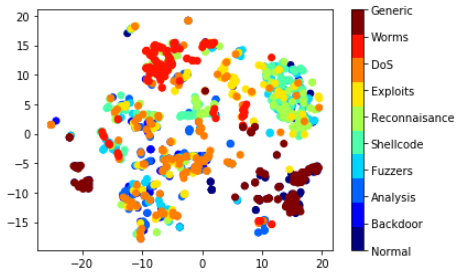

Figure 3: The t-SNE scatter plot of the UNSW-NB15

C. Experiment Results

Accuracy, as a performance measure was deployed to measure the performance and effectiveness of the proposed Generator network and the original dataset, UNSW-NB15. This performance measures is computed by using the following equation:

Accuracy = (TP+TN) / (TP+FP+TN+FN)

Where: TP represents True Positive meaning the correct intrusion detection.

TN represents True Negative denoting a normal traffic.

FP representing False Positives which means that detections which assume normal traffic as attacks.

FN representing False Negatives meaning it has failed to disclose an intrusion.

The following table 3. was showing a comparison of the accuracy rate of our proposed framework with different machine learning methods such as CNN, BP neural network and the Support Vector Machine (SVM) on the UNSW-NB15 Dataset. Our algorithm has shown to have an

accuracy of 85.40% unlike SVM, BP Neural Network and CNN each with an accuracy of 65.73%, 62.60% and 63.55% respectively. The generated data with an accuracy of 85.4 which is obtained from the generator network is fed into the generator for the second time to generate new data.

This way, the Generator network is able to generate a better quality data taking much less time than the initial training time.

Method	Accuracy
CNN	63.55%
BP Neural network	62.60%
Support Vector Machines	65.73%
Proposed Approach	85.40%

TABLE 3. results of different algorithms

V. CONCLUSION

In this paper, we research on Wireless Sensor Networks and the use of Reinforcement Learning and Generative Adversarial Networks to solve the problem of intrusion detection as the currently existing middlewares do not completely address the problems which impact significantly Wireless Sensor Networks performance. The proposed algorithm consists of a GAN with two neural networks, a Generator (G) and a Discriminator (D). The Generator is able to generate fake data which has a data distribution similar to that of the samples its been trained on. Its main goal is to trick the discriminator. This makes it harder for an attacker to be able to differentiate between the two data samples. Reinforcement Learning is utilized to control the complexity of the latent space in Generative Adversarial Networks. To the best of our knowledge, we are the first to introduce this unique combination of Reinforcement Learning (RL) and Generative Adversarial Network (GAN) where an RL agent controls a GAN in solving the intrusion detection problem in Wireless Sensor Networks (WSN). Thus, the use of an RL agent in our proposal replaces the need for performing complex optimization making our technique both real time and efficient. In the future, the proposed data quality algorithm improvement will be applied to a large scale wireless sensor networks. It is also important to note that this is still a research idea that needs to be further researched on. Still researching on ways machine learning can be used intelligently to dynamically allocate the data queue threshold. Wireless Sensor Network performance data can be collected based on its performance in different speeds, train the network using this data and evaluate the performance of the model on new data from the nodes.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Jun Tu for the support after a whole year of reading, researching, developing, testing and finally, being able to write my this paper.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] E. H. Callaway Jr, *Wireless sensor networks: architectures and protocols*. CRC press, 2003.

[3] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, “A review on generative adversarial networks: Algorithms, theory, and applications,” *arXiv preprint arXiv:2001.06937*, 2020.

[4] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, “Generative adversarial network for wireless signal spoofing,” in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, 2019, pp. 55–60.

[5] H. Chen and L. Jiang, “Gan-based method for cyberintrusion detection,” *arXiv preprint* arxiv:1904.02426, 2019.

[6] Zenati, H. , et al. "Efficient GAN-Based Anomaly Detection." *arXiv preprint arXiv:* 1904.02426 .2019

[7] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, 2020.

[8] P. G. Om Prakash, B. Maram, G. Nalinipriya, and R. Cristin, "Harmony search Hawks optimization-based Deep reinforcement learning for intrusion detection in IoT using nonnegative matrix factorization," *International Journal of Wavelets Multiresolution and Information Processing*, vol. 19, no. 04, Jul 2021, Art. no. 2050093.

[9] B. Z. Wang, Y. Sun, M. Y. Sun, and X. D. Xu, "Game-Theoretic Actor-Critic-Based Intrusion Response Scheme (GTAC-IRS) for Wireless SDN-Based IoT Networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1830-1845, Feb 2021.

[10] M. S. Frikha, S. M. Gammar, A. Lahmadi, and L. Andrey, "Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey," *Computer Communications*, vol. 178, pp. 98-113, Oct 1 2021.

[11] C. Kim and J. Park, "Designing online network intrusion detection using deep auto-encoder Q-learning," *Computers & Electrical Engineering*, vol. 79, Oct 2019, Art. no. 106460.

[12] D. Pfau and O. Vinyals, “Connecting generative adversarial ne -tworks and actor-critic methods,” *arXiv preprint arXiv:1610.01 945, 2016*.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.

[14] N. Ketkar, “Introduction to keras,” *Deep Learning with Python. S pringer*, 2017, pp. 97–111.

[15] Moustafa, N. , and J. Slay . "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." In 2015 *Military Communications and Information Systems Conference (MilCIS)*, pp.1-6 IEEE, 2015.

[16] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp.2579–2605, 2008.