A Project Report

On

Reinforcement Learning in Cybersecurity

BY

Pratyush Bindal

2022A7PS0119H

Under the supervision of

Dr. Paresh Saxena

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF

CS F266: STUDY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(May 2024)

ACKNOWLEDGMENTS

I want to express my sincere gratitude to Dr. Paresh Saxena for all of his guidance and support during this research.

I would also like to express my sincere gratitude to Ms. Lalitha Chavali for her valuable time and work on the project, without whom this study would remain incomplete. Her insightful feedback, guidance and meaningful suggestions proved to be very helpful throughout the project.

Pratyush Bindal

(2022A7PS0119H)

Birla Institute of Technology and Science-

Pilani,Hyderabad Campus

Certificate

This is to certify that the project report entitled "**Reinforcement Learning in Cybersecurity**" submitted by Mr. Pratyush Bindal (ID No. 2022A7PS0119H) in partial fulfilment of the requirements of the course CS F266, Study Project Course, embodies the work done by him under my supervision and guidance.

Date: May 10, 2024 (**Dr. Paresh Saxena**)

BITS- Pilani, Hyderabad Campus

ABSTRACT

Network security dramatically depends on network intrusion detection. Even though the most recent deep learning-based intrusion detection algorithms have demonstrated strong detection performance, handling unbalanced datasets and recognising unknown and minority threats remain challenges. There are still limitations in dealing with unbalanced datasets and identifying minority attacks and unknown attacks.

The study details the development and use of an intrusion detection system's AE-SAC model based on an adversarial environment learning and soft actor-critic reinforcement learning algorithm as introduced in paper (i). The study also outlines creating and using an intrusion detection system built on the Twin Delayed DDPG (TD3) Model introduced in paper (iv). This off-policy method combines target policy smoothing, delayed updating of policy & target networks, and clipped double Q-learning.

This study first addresses the imbalance issue with the original data by using an environmental agent for training data resampling. Second, reinforcement learning redefines rewards. Hence, different reward levels were established for different types of attacks to increase the recognition rate of a few categories of network attacks.

CONTENTS

INTRODUCTION

Intrusion Detection technology (IDS) produces a resilient defence system in real-time to prevent, transfer, and lessen information systems' risks by using an active defence approach, allowing for early and accurate warning before incursions negatively affect computer systems. Machine learning techniques can solve intrusion detection problems because of the abundance of network telemetry and other security data.

Deep reinforcement learning (DRL) can solve numerous complex practical problems by fitting neural networks to the Markov Decision Process (MDP), which has been developed recently, combining deep learning and reinforcement learning. The advent of DRL has provided a fresh method for addressing the intrusion detection issue.

Soft Actor-Critic (SAC) is regarded as one of the most effective reinforcement learning methods introduced and explored in the paper (ii) and (iii). The method uses a maximum entropy reinforcement learning model to determine the best course of action that maximises long-term entropy and predicts rewards.

Another effective reinforcement learning technique that has gained popularity for network intrusion detection is Twin Delayed Deep Deterministic Policy Gradient (TD3), which has its roots in the paper (iv). It provides efficiency and robustness while learning policies for continuous action spaces.

After carefully analysing previous research studies and documentation, we applied the AE-SAC and TD3 Models in this study project. We evaluated how effective they were against intrusion attempts using the NSL-KDD dataset. Additionally, we reduced Type I and II mistakes and enhanced both models' capacity to detect network intrusions after varying the parameters and running them several times.

The susceptibility of neural network models to adversarial attacks is a crucial concern that needs attention. However, we plan to utilise adversarial training to increase the model's resilience in the future.

Our effort, in summary, highlights the potential of SAC and TD3-based reinforcement learning to strengthen intrusion detection systems against changing cyber threats. Our goal is to develop a robust defence framework that can protect information systems with increased accuracy and resilience by using adversarial training mechanisms.

OBJECTIVES

1. Implementation of AE-SAC and TD3 Model on NSL-KDD Dataset for Intrusion Detection:

- Implemented AE-SAC and Twin Delayed DDPG (TD3) Algorithms on NSL-KDD Dataset for Intrusion Detection
- Modified, trained and evaluated the AE-SAC and TD3 models for intrusion detection focusing on its accuracy & efficiency in identifying network intrusions.

2. Utilisation of MQTT, NSL-KDD and AWID Dataset for Intrusion Detection:

- Employed various types of classifiers including XGBoost, CatBoost, LightGBM, Decision Trees, and KNN on the datasets for intrusion detection purposes.
- Compared the effectiveness of these classifiers in detecting network intrusions and comprehensively evaluate their performance metrics.

3. Developing ML-Based Intrusion Detection Algorithm on NSL-KDD Dataset

- Established ML-based Intrusion Detection System on NSL-KDD Dataset to map alert-attack relations
- Performed anomaly detection for network intrusion by training multiple classifiers including Logistic Regression, SVM, KNN, Decision Trees, Random Forests, & SGD Classifier with various strategies, generating alerts using Gaussian Mixture Models (GMM) clustering, and training binary classifiers for each alert type to distinguish true & false positives.

4. Evaluation and Comparison of Detection Methods:

- Evaluated performance metrics such as precision, recall, and F1-score to assess the effectiveness of detection methods.
- Compared detection methods in terms of complexity, scalability, and robustness.
- Optimised the code and explored adversarial training techniques to enhance the model's durability and resilience against attacks.

COURSE OF STUDY

First, we learned about the basic ideas of Reinforcement Learning (RL), such as reward, action, state, and policies. By examining value functions, reward discounting, and the Markov Decision Process (MDP), we could understand how reinforcement learning (RL) agents learn to maximise cumulative rewards. We studied a variety of reinforcement learning algorithms, including Q-learning, Actor-Critic models, Monte Carlo approaches, and dynamic programming strategies.

A thorough examination of the papers (ii) and (iii) led us to focus on the Soft Actor-Critic (SAC) model when we looked at policy optimisation techniques. In order to enhance the model for intrusion detection, we read paper (i). This paper introduced the Environmental Agent and Classifier Agent framework, emphasising the design of reward functions specific for intrusion detection tasks.

We read paper (iv) and learned about clipped Q-learning techniques, target networks, and delayed policy updates specific to TD3 algorithm to avoid overestimation bias of the implemented actor-critic methods. Error Accumulation Correction and Target Policy Regularisation were also investigated to enhance the model's performance.

Additionally, we used the MQTT, AWID and NSL-KDD datasets, serving as a baseline for network intrusion detection. We utilised a suite of classifiers, such as LightGBM, XGBoost, Support Vector Machines (SVM), and CatBoost, using the acquired knowledge. We compared how well these classifiers detected network intrusions and assessed their performance indicators thoroughly.

We focused much of our attention on classifier training techniques, such as hyper-parameter tuning and feature extraction, which are essential for improving intrusion detection system performance. We carefully experimented with investigating a variety of machine learning (ML) approaches, including Linear SVC, Random Forest Classifier, One-Vs-All (OVA) Multi-Class Classification, and ensemble methods, including Boosting and Bagging. Furthermore, to address the problem of class imbalance that frequently arises in intrusion detection datasets, we employed a variety of data sampling approaches, such as Random Oversampling and Synthetic Minority Over-sampling Technique (SMOTE). Furthermore, we used the Gaussian Mixture Model (GMM) to detect fraud on the NSL-KDD dataset and identify unusual trends.

METHOD FOR NETWORK INTRUSION DETECTION

According to paper (i), we employed the AE-SAC model. The AE-SAC model incorporates an environmental agent for data sampling, and both the environment agent and the classifier agent are implemented using the SAC architecture. We implemented the SAC model by studying paper (ii).

Five different neural network architectures are being used-

1. An Actor network that determines the probability distribution of the actions
2. Two V Critic Networks for State Value Estimation (One is the Evaluation Network, and the other is the Target Network, which is updated via Soft Update).
3. Two Q Critic Networks: Calculating State Action Values (trained with the MSE Loss Function and the same update procedure)

The paper introduces Experience Replay into the SAC model, a replay memory technique used in RL where we store the Agent's experiences at each time in a data set, pooled over many episodes into a replay memory. We then sample the memory randomly for a mini-batch of experience and use this to learn off-policy, leading to experience data having a temporal dependency.

Environmental Agent is added to the Original RL Framework to address the dataset imbalance issue by dynamically resampling the dataset during training. Initially, the environment agent is given random training data. As training proceeds, the environment agent learns and samples the training data that maximises its reward.

The Classifier Agent acts as the Actor and the Environmental Agent as the Critic, which is rewarded when the classifier fails to recognise; otherwise, the classifier is rewarded. The environment and classifier agents perform adversarial learning around obtaining the maximum reward. The environment agent gets the maximum reward by sampling more data, which the classifier cannot recognise or classify incorrectly. Conversely, classifiers receive a larger reward by recognising or correctly classifying more data. Both classifier and environmental agents are trained in parallel.

We used the NSL-KDD dataset, which was pre-processed, including data cleaning (replaced infinity value with -1 and removed NULL rows), conversion (used One Hot Encoding to map non-numeric features to numeric features) and normalisation (used Max-Min Normalisation Method).

After meticulously studying the paper (iv), we employed the TD3 algorithm for network intrusion detection on the NSL-KDD dataset. The TD3 algorithm is an

extension of the original Deep Deterministic Policy Gradient (DDPG) algorithm introduced in the paper (v), designed to improve stability and performance in continuous action spaces.

Many RL-based algorithms are unstable and rely profoundly on their hyper-parameters due to continuously overestimating the Q values of the Critic network. As a result of these cumulative estimating mistakes, the agent may eventually undergo catastrophic forgetfulness or slip into a local optima. In order to solve this problem, TD3 concentrates on lessening the overestimation bias for Actor-Critic Methods. This is done by implementing three key features-

1. Using two separate Critic networks and using the smallest value of two during target formation
2. It makes training efficient by delaying updates on the actor-network. It updates the Actor every two-time steps instead of after each time step. This delay helps to stabilise training and mitigates the risk of overfitting.
3. TD3 incorporates target policy smoothing to reduce overestimation bias. This involves adding noise to the target action during the critic update step.

The TD3 algorithm also incorporates Experience Replay to enhance sampling efficiency and stabilise training. Experiences are randomly taken from a replay buffer to train the networks off-policy. Furthermore, an environmental agent can handle dataset imbalance by dynamically resampling data. Rewards are maximised by adversarial learning between the environmental agent and the classifier.

First, we initialise Actor and Critic neural networks and a Replay Buffer for intrusion detection. Transitions are saved in the replay buffer, and actions are selected using actor networks with exploration noise. Iterative changes are made to the networks, with actor networks updated after critic networks. For stability, target networks are updated regularly. This procedure continues until the reward function is maximised. Due to this iterative learning, TD3 can adjust and improve intrusion detection efficiency over time on the NSL-KDD dataset.

Data preparation is carried out as we did while implementing the SAC model. Using the NSL-KDD dataset, we cleaned the data (removing NULL rows and replacing infinity values with -1), converted the non-numeric features to numeric features using One Hot Encoding, and normalised the results using Max-Min Normalisation.

EVALUATION AND CONCLUSION

We evaluated both AE-SAC and TD3 models based on various parameters such as Accuracy, TPR (True Positive Rate), FPR (False Positive Rate), MCC (Matthews Coefficient) and F1-Score. Both the algorithms were rigorously studied and it was found out that TD3 performed much better than the AE-SAC model due to its key features of double Q-Clipped Learning and Target Policy Smoothing. Various modifications to enhance the efficacy of TD3 model for intrusion detection were developed:

- Data Preprocessing: Standard Scaler was used to guarantee standardised inputs.
- Activation Functions: ReLU, Leaky ReLU, ELU, SoftPlus, Sigmoid, and Swish were tried as activation functions.
- Architecture: Changes to the number of linear layers were investigated although they did not lead to significant performance improvements
- Loss Functions and Optimisers: Along with MSE, Binary CrossEntropy, and Hinge loss functions, the following optimisers were tested: Adam, SGD, Adagrad, AdamW, and RMSProp.
- Replay Buffer: Used a Replay Buffer to store and sample experiences to improve the consistency and effectiveness of learning.
- Update Timing: Modifications were made to account for periodic policy updates.
- Hyper-parameter adjustments were made to the discount factor, learning rate, and exploration noise.

These experiments provide insight on how to optimise the TD3 model's performance and resilience for anomaly detection uses.

<u>Results for AE-SAC and TD3 implementation on NSL-KDD Dataset:</u>

Mean F1 Score = 0.748
Mean Accuracy = 0.842

Mean F1 Score = 0.649
Mean Accuracy = 0.745

TD3_NSL_KDD

Accuracy	TPR	FPR	MCC	F1
0.830	0.796	0.058	0.781	0.739
0.879	0.841	0.077	0.764	0.758
0.818	0.823	0.063	0.761	0.747

SAC_NSL_KDD

Accuracy	TPR	FPR	MCC	F1
0.740	0.557	0.044	0.505	0.642
0.748	0.582	0.056	0.512	0.653
0.747	0.577	0.051	0.511	0.652

Furthermore, we utilised a variety of classifiers including XGBoost, CatBoost, LightGBM, Decision Trees, and KNN and carefully compared their efficacy in detecting network intrusions by utilising the NSL-KDD, MQTT and AWID dataset enabling us to gain a thorough understanding and mathematical understanding of various ML-based methods through the reading of numerous research papers. Additionally, we thoroughly examined the datasets to determine the best practices for identifying abnormalities and fraudulent activity in each dataset.

Results for various Classifiers used for Fraud and Anomaly Detection on MQTT, AWID and NSL-KDD Dataset:

Summary XGBoost

	precision	recall	f1-score	support
0	0.88	0.20	0.33	3879.00
1	0.86	0.84	0.85	3340.00
2	0.94	0.92	0.93	3378.00
3	0.76	1.00	0.86	5967.00
4	0.64	0.83	0.72	4352.00
accuracy	0.78	0.78	0.78	0.78
macro avg	0.82	0.76	0.74	20916.00
weighted avg	0.80	0.78	0.74	20916.00

Summary CatBoost

	precision	recall	f1-score	support
0	0.96	0.98	0.97	3879.00
1	1.00	1.00	1.00	3340.00
2	1.00	1.00	1.00	3378.00
3	1.00	1.00	1.00	5967.00
4	0.99	0.96	0.97	4352.00
accuracy	0.99	0.99	0.99	0.99
macro avg	0.99	0.99	0.99	20916.00
weighted avg	0.99	0.99	0.99	20916.00

Summary LightGBM

	precision	recall	f1-score	support
0	0.59	0.60	0.59	3879.00
1	0.79	0.88	0.83	3340.00
2	1.00	0.98	0.99	3378.00
3	0.97	0.99	0.98	5967.00
4	0.67	0.60	0.63	4352.00
accuracy	0.82	0.82	0.82	0.82
macro avg	0.80	0.81	0.81	20916.00
weighted avg	0.81	0.82	0.81	20916.00

Summary Decision Trees

	precision	recall	f1-score	support
0	0.97	1.00	0.98	3879.00
1	0.99	0.88	0.93	3340.00
2	1.00	0.99	0.99	3378.00
3	1.00	1.00	1.00	5967.00
4	0.92	0.97	0.94	4353.00
accuracy	0.97	0.97	0.97	0.97
macro avg	0.97	0.97	0.97	20917.00
weighted avg	0.97	0.97	0.97	20917.00

Summary KNN

	precision	recall	f1-score	support
0	0.96	0.99	0.98	3879.00
1	1.00	1.00	1.00	3340.00
2	1.00	1.00	1.00	3378.00
3	1.00	1.00	1.00	5967.00
4	0.99	0.96	0.98	4353.00
accuracy	0.99	0.99	0.99	0.99
macro avg	0.99	0.99	0.99	20917.00
weighted avg	0.99	0.99	0.99	20917.00

Summary Linear

	precision	recall	f1-score	support
0	0.68	0.96	0.80	3879.00
1	0.99	0.22	0.36	3340.00
2	0.97	0.97	0.97	3378.00
3	0.98	1.00	0.99	5967.00
4	0.83	1.00	0.90	4353.00
accuracy	0.86	0.86	0.86	0.86
macro avg	0.89	0.83	0.80	20917.00
weighted avg	0.89	0.86	0.83	20917.00

Result for ML-based anomaly detection method on NSL-KDD Dataset:

The NSL-KDD dataset underwent another evaluation for ML-based anomaly detection method, employing diverse classifiers, Gaussian Mixture Models for alert generation, and binary classifiers for discerning true and false positives. Assessment based on precision, recall, and F1-score indicated mixed performance across alert types. Overall, the evaluation highlighted the method's effectiveness in detecting certain types of anomalies while identifying areas for improvement in others. The evaluation of the anomaly detection model on the NSL-KDD dataset yielded the following results:

Summary

Alert Type	Normal	DoS	Probe	R2L	U2R
Precision	0.85	0.92	0.96	0.89	0.88
Recall	0.89	0.94	0.97	0.90	0.91
F1-Score	0.94	0.97	0.99	0.95	0.95
Support Instances	438	438	438	438	438
Average Duration	2056.07	35324.5	24647.67	10999.0	39899.5
Number of Attacks	11648	2	3	1	2

Overall, this effort marks a substantial advancement in intrusion detection technologies. We hope to contribute to create a more robust and dependable intrusion detection systems by employing novel methodologies and conducting extensive evaluations, thereby supporting cybersecurity efforts in an ever-changing digital ecosystem.

REFERENCES

(i) Li, Z., Huang, C., Deng, S., Qiu, W., & Gao, X. (2023, December). A soft actor-critic reinforcement learning algorithm for network intrusion detection. *Computers & Security*, *135*, 103502. https://doi.org/10.1016/j.cose.2023.103502

(ii) Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, January 4). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv.org. https://doi.org/10.48550/arXiv.1801.01290

(iii) Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu,H., Gupta, A., Abbeel, P., & Levine, S. (2018, December 13). *Soft Actor-Critic Algorithms and Applications*. arXiv.org. https://doi.org/10.48550/arXiv.1812.05905

(iv) Fujimoto, S., Herke, V. H., & Meger, D. (2018, February 26). Addressing Function Approximation Error in Actor-Critic Methods. arXiv.org. https://doi.org/10.48550/arXiv.1802.09477

(v) Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015, September 9). *Continuous control with deep reinforcement learning*. arXiv.org. https://doi.org/10.48550/arXiv.1509.02971

(vi) *Deep Reinforcement Learning for Cyber Security*. (2023, August 1). IEEE Journals & Magazine | IEEE Xplore. https://doi.org/10.1109/TNNLS.2021.3121870

(vii) VKumar, V. (2021, December 7). *Soft Actor-Critic Demystified - Towards Data Science*. Medium. https://towardsdatascience.com/soft-actor-critic-demystified-b8427df61665

(viii) Verma, N. (2023, November 8). *Overview of Classification Algorithms in Machine Learning*. Medium. https://medium.com/@nandiniverma78988/overview-of- classification-algorithms-in-machine-learning-3050d9e14c5c

(ix) Wong, K. J. (2022, November 29). *CatBoost vs. LightGBM vs. XGBoost - Towards Data Science*. Medium. https://towardsdatascience.com/catboost-vs-lightgbm-vs- xgboost-c80f40662924

(x) Byrne, D. (2023, September 12). *TD3: Learning To Run With AI - Towards Data Science*. Medium. https://towardsdatascience.com/td3-learning-to-run-with-ai-40dfc512f93