HDRL-IDS: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Enhancing the Security of Medical Applications in 5G Networks

Ali Ghubaish
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO
aghubaish@wustl.edu

Zebo Yang
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO
zebo@wustl.edu

Raj Jain
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO
jain@wustl.edu

***Abstract*— The evolution of next-generation wireless networks, particularly the integration of Multi-access Edge Computing (MEC) in 5G, is set to revolutionize the infrastructure of secure systems. This is exemplified on the Internet of Medical Things (IoMT) field, where benefits such as remote surgeries and diagnostics become increasingly common, especially in pandemic scenarios. However, incorporating MEC services into the 5G framework significantly enlarges the network's vulnerability to traditional security breaches and introduces new, sophisticated types of attacks. These emerging threats, often undetectable by conventional methods, necessitate the development of an adaptive Intrusion Detection System (IDS) capable of identifying such complex security challenges with minimal human intervention. To tackle these issues, this paper introduces a novel Hybrid Deep Reinforcement Learning IDS (HDRL-IDS), rooted in an actor-critic methodology, designed to detect more complex security threats adaptively and autonomously with limited human oversight. Our HDRL-IDS combines the analysis of both network and host features, effectively leveraging the advantages of both Network IDS (NIDS) and Host IDS (HIDS). Empirical results indicate that our HDRL-IDS outperforms traditional NIDS and HIDS in threat detection efficacy. Furthermore, we present a novel dataset derived from an emulated 5G testbed with integrated MEC services to facilitate advanced research and development in the field, specifically for applications designed to address intricate attacks and scenarios demanding high reliability.**

Keywords—Deep Reinforcement Learning, Intrusion Detection Systems, Mobile Edge Clouds, Internet of Medical Things, 5G, Deep Deterministic Policy Gradient.

I. INTRODUCTION

Advancements in next-generation wireless networks, particularly in healthcare applications like the Internet of Medical Things (IoMT) and mobile healthcare, are set to bring transformative changes. The 5G infrastructure, for instance, offers substantial connection capacity and faster speeds, making it particularly advantageous for high-risk operations such as remote surgery [1]. Ultra-Reliable Low Latency Communications (URLLC) services make this possible in remote hospitals during pandemics, which provide exceptional data rates. Additionally, 5G incorporates Multi-access Edge Computing (MEC) to enhance low-latency applications by processing data closer to the network edge.

In 5G infrastructure, edge clouds or MECs are close to the User Equipment (UE), typically at cellular towers. However, due to its distributed nature, the edge cloud is more vulnerable to attacks than traditional, centralized data center infrastructures. Given the evolution of attack techniques aimed at bypassing Intrusion Detection Systems (IDS) and the expanded security surface associated with MEC integration, there is a pressing need for a new generation of IDS. These systems must be capable of intelligently evolving to detect new attacks with minimal human interaction [2].

Deep Reinforcement Learning (DRL) has emerged as a promising candidate for such dynamic IDS, offering solutions with little need for human oversight to detect new attacks (zero-day attacks) without retraining the IDS model. DRL combines the strengths of Reinforcement Learning (RL) and Deep Learning (DL), allowing the RL agent to learn effectively from a low-dimensional feature space [2, 3]. To address the increasing prevalence of zero-day attacks, the primary focus of DRL for IDS has been on Network IDS (NIDS). However, these methods often rely on centralized architectures, which are not readily applicable to the MEC services in 5G.

Most studies prioritize NIDS over Host-based IDS (HIDS) because of the suitability of NIDS for implementing RL algorithms like Q-Learning and Deep Q-Learning (DQL). These methods effectively analyze and learn from network traffic patterns but can be biased and inefficient for large states or action spaces [4]. Additionally, many studies rely on datasets that do not adequately reflect the characteristics of 5G networks, with the NSL-KDD dataset being a common example [5].

To address these gaps, we integrate the advantages of both NIDS and HIDS within an actor-critic DRL framework. We utilize the Deep Deterministic Policy Gradient (DDPG), known for its model-free and online capabilities. Moreover, we have developed an emulated 5G testbed with MEC services designed explicitly for healthcare systems. The testbed produced a dataset that more closely aligns with the

ePrint

2024 International Conference on Smart Applications, Communications and Networking (SmartNets)
979-8-3503-8532-8/24/$31.00 ©2024 IEEE

most recent developments in 5G network technology. Built using Simu5G [6], our testbed replicates a 5G network environment with multiple MEC servers and UEs based on the OMNet++ framework [7]. Overall, the key contributions of this paper are as follows:

1. We design a replicable emulated 5G testbed with MEC service for healthcare systems, enhancing IDS research in this field.

2. To address the gaps in using NIDS and HIDS, we introduce a hybrid DRL model for IDS in 5G MEC that combines host and network features based on actor-critic techniques.

3. We compare our hybrid DRL IDS model against traditional NIDS and HIDS and find that our model outperforms them in threat detection efficacy.

The rest of the paper is organized as follows. Section II provides a background on the DRL model and 5G infrastructure. Sections III and IV discuss related work and our proposed HDRL testbed. Section V presents our testbed results. Conclusions, and future work are detailed in Section VI.

II. BACKGROUND

This section introduces the typical DRL algorithms and the 5G infrastructure to provide readers with fundamental insights into our approaches.

A. DRL

RL is a subset of Machine Learning (ML) techniques in which an agent learns decision-making by engaging with its environment, akin to how humans acquire knowledge through trial and error. RL comprises five key components: agent, environment, state, action, and reward. The agent is the decision-maker, while the environment is what the agent interacts with. The state represents the current situation in the environment. The action is the agent's decision based on the state, determining whether the agent receives a reward for a correct action or a penalty for an incorrect one.

DRL extends RL by integrating Deep Neural Networks (DNN) to handle complex, high-dimensional environments. This has led to an increased interest in DRL in recent years. The main categories of RL are as follows (as shown in Fig. 1 [8]):

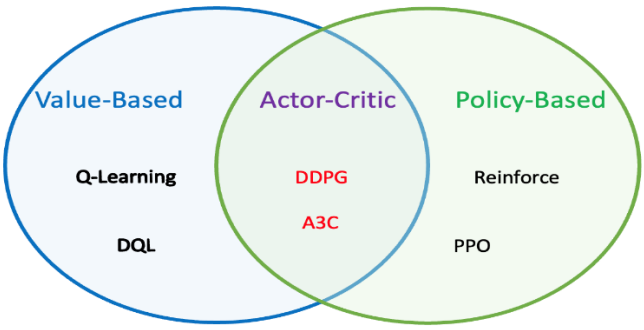


Fig. 1. Classifications of RL categories.

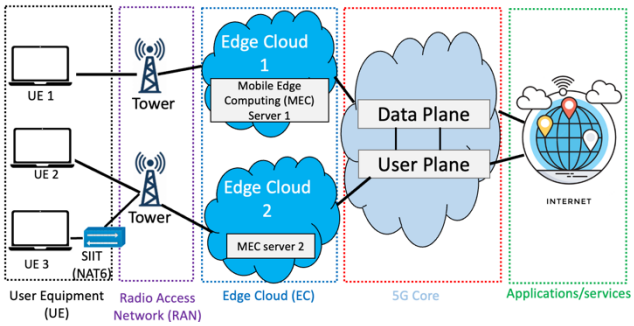


Fig. 2. 5G End-to-End System Infrastructure.

1) **Value-based methods:** These methods aim to estimate the value of actions in the current state to find the optimal action that maximizes rewards. Examples include Q-learning and DQL. However, they are prone to high bias [9].

2) **Policy-based methods:** Instead of estimating action values, these methods directly learn a policy that maps states to actions. Examples are Reinforce and Proximal Policy Optimization (PPO). However, they can suffer from high variance and large gradient noise [9].

3) **Actor-critic methods:** Combining the strengths of value-based and policy-based methods, they consist of an actor which makes decisions and a critic which evaluates these decisions to help the actor adjust its policy for better performance. Examples include DDPG and Asynchronous Advantage Actor Critic (A3C).

In this paper, we employ DDPG, an actor-critic method, for our HDRL model.

B. 5G Infrastructure

Fig 2 illustrates the system-level components of the 5G infrastructure, including End Users, the Radio Access Network (RAN), the Edge Cloud, and the Core Cloud. We focus on the URLLC applications, for which the Edge Cloud is a crucial enabler. UEs include two types of users: direct users and indirect users. Direct users are directly connected to the 5G network, while indirect users are inside a private network to benefit from 5G services while not being directly connected to the 5G network. All UEs significantly benefit from MECs in 5G networks by experiencing lower latency and faster data processing. MECs allow data processing closer to the data source at the network's edge.

While 5G has overcome many limitations of 4G networks, it has an increased attack surface. The 5G infrastructure is susceptible to various attack vectors, such as Distributed Denial of Service (DDoS) attacks targeting the edge cloud's network infrastructure. Other potential threats include ransomware, buffer overflow attacks, and Man-in-The-Middle (MiTM) attacks, which can involve data manipulation during transmission.

III. LITERATURE REVIEW

IDS are pivotal in network security and are classified primarily into two types: Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS) [10]. SIDSs are highly effective

in recognizing known attacks leveraging predefined patterns. However, they fall short in identifying novel, unseen threats. AIDSs, conversely, offer a solution to this limitation by focusing on deviations from normative behavior, making them apt for detecting zero-day attacks. Zero-day attacks, such as the infamous Stuxnet worm [11], which targeted industrial control systems, are cyber threats that exploit previously unknown software or hardware vulnerabilities before developers can issue a fix or patch. Consequently, AIDSs have seen increasing focus in recent literature, particularly those employing DRL.

Among the notable works, Benaddi et al. [1] proposed a DRL-IDS specifically for the Internet of Things (IoT) and Wireless Sensor Networks (WSN), utilizing a DQL model. Their system demonstrated remarkable detection rates and low false negatives when benchmarked against standard RL and K-Nearest Neighbors (KNN) using the NSL-KDD dataset.

Similarly, Hsu and Matsuoka [2] designed a DRL-based IDS aligned with Benaddi et al.'s approach. Still, they extended their comparison to include machine learning models like Random Forest (RF) and Support Vector Machine (SVM), using both NSL-KDD and UNSW-NB15 datasets. Their results indicated superior accuracy of the DRL-based IDS in detecting intrusions across these datasets.

Expanding on these foundations, Alavizadeh et al. [12] developed a DQL-based IDS, focusing on efficiency with a training duration of just 250 episodes. The model achieved a notable accuracy rate, highlighting the effectiveness of DRL in IDS.

Ren et al. [13] introduced an innovative approach with their ID-RDRL, a NIDS that combines DQL with recursive feature elimination (RFE) to reduce redundant features significantly. This strategy not only streamlined the feature set but also enhanced the NIDS's performance, as evidenced by their accuracy and F_1 scores of 96.20% and 94.90%, respectively, on the CSE-CIC-IDS2018 dataset.

In the context of 5G networks, Moudoud and Cherkaoui [5] constructed a DRL-based IDS to enhance the security of IoT systems. Utilizing a DQL model and training on the NSL-KDD dataset, their system showed promising results, achieving a high F_1 score of 94% with limited training episodes.

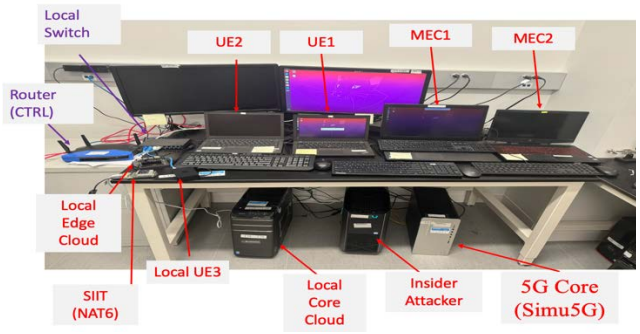


Fig. 3. Our Testbed.

Our research diverges from these existing studies by introducing a unique dataset tailored for 5G networks with MEC services, employing hybrid data sources from both network and host and leveraging an actor-critic DRL method. This approach combines the advantages of both NIDS and HIDS, making it a novel contribution to the field of anomaly-based, DRL-based IDS. To the best of our knowledge, this is the first instance of an IDS that integrates these specific features for enhanced intrusion detection in 5G networks. Additionally, it marks the first dataset generated by an emulated 5G testbed with MEC services.

IV. HYBRID DEEP REINFORCEMENT LEARNING (HDLR) TESTBED

HDRL consists of IDS distributed components in the edge clouds (i.e., MEC) of the 5G network to provide IDS for both direct and indirect users. In other words, an HDRL component is integrated into every edge cloud, allowing for the dynamic updating of IDS models that combine the benefits of using host data and network-based IDS in each edge cloud. Furthermore, using an IDS in the edge clouds eliminates the single point of failure issues in centralized IDSs, minimizes the effects of an attack on the whole system, and increases the system's resilience.

The IDS can be enhanced and adapted to different types of attacks (including zero-day attacks) with the help of DRL algorithms with two innovations. DRL enhances and adapts IDS for new attacks by learning and updating policies from environmental interactions. Hence, it enables continuous adaptation to emerging threats without the extensive retraining required by other ML and DNN methods. First, to avoid the drawbacks of value-based and policy-based algorithms [9], we use one of the actor-critic algorithms, DDPG. Second, to take advantage of both types of IDS, we create a hybrid DRL model in the MECs. We combine the advantages of host data and network-based IDS.

We have developed a testbed to emulate user and attacker interactions to the 5G MEC and core. First, the edge clouds monitor and collect the network metadata for the links between the edge clouds and the edge nodes. Next, the UEs send the host metadata (i.e., CPU usage and running processes) to the edge clouds as part of the transmitted data used by the hybrid DRL model. Finally, our algorithm executes the IDS procedures, processing the network and host metadata in combination as a hybrid approach.

A. Testbed Architecture

The testbed comprises six main components, as depicted in Fig. 3, based on the structure in Fig. 2: a 5G core, a local network, two MECs, three UEs, an insider attacker, and a router.

1) **5G Core:** At the core of the testbed, the Ubuntu 20.04 computer emulates the behaviors of the 5G infrastructure using Simu5G. The emulation setup is based on the "extUeAppMecApp" instance from Simu5G v1.2.2, initially facilitating the emulation of one MEC and one UE. We have extended the system to include multiple MECs and external hosts, providing a more complex and representative

emulation of a 5G network. The core host is the command center for all communications within the testbed's 5G infrastructure.

2) **Local Network:** This component replicates a typical local network linked to the broader 5G network. It includes edge clouds with a local core, a local switch, and a Stateless IP/ICMP Translation (SIIT) system implemented using the Jool package [14]. The SIIT component, particularly crucial, addresses the Network Address Translation (NAT) issues, enabling local devices to use IPv6 for individual identification within OMNET-emulated 5G networks, which predominantly utilize IPv4. Given the current limitations of OMNet++ version 6.0 in terms of IPv6 support, this solution is a workaround, which is necessary for complete compatibility with Simu5G.

3) **MECs:** Running on Ubuntu 20.04 OS, the PCs hosting the MEC services are integral to the testbed. Here, the HDRL IDS model is stationed, monitoring network data via Argus [15] and gathering host data using the psutil library [16]. The collected data is crucial for the functioning the hybrid IDS model within the testbed.

4) **UEs:** UE1 and UE2, based on Ubuntu 20.04 OS, are directly connected to the 5G network. In contrast, UE3, operating on Raspberry Pi OS, is situated within the local network, representing a diverse range of end-user scenarios.

5) **Insider Attacker:** This Ubuntu 20.04 computer is configured to generate various attacks, including MiTM, DDoS, ransomware, and buffer overflow attacks, as elaborated in Section IV.B. This component is essential for testing the resilience and effectiveness of the IDS against a range of security threats.

6) **Router:** The router's role is to manage the entire testbed from a central control PC. Its function is critical in ensuring unbiased data collection for the new dataset. Randomizing the timing and duration of attacks helps create a diverse and unpredictable set of scenarios for robust IDS testing.

Each component creates a dynamic and realistic 5G network environment, enabling thorough testing and evaluation of the proposed HDRL IDS model in various scenarios and attack conditions.

B. Types of Attacks

In this dataset, four types of attacks are implemented, offering a broad spectrum of applications in various domains. All attacks are developed using Python, except for the buffer overflow attack, which is executed using a Bash script. The descriptions of these attacks are as follows:

1) **MiTM:** This attack involves data alteration, where host data sent from an UE to a MEC server is altered. It primarily targets UE1 and UE3. The Scapy [17] and NetfilterQueue [18] libraries intercept and modify the packets before redirecting them to the MEC server.

2) **DDoS:** Executed by multiple compromised PCs, this attack floods MEC1 and MEC2 with overwhelming requests, disrupting services for regular users like UE1, UE2, and UE3. The attacker's PC simulates a DDoS attack from multiple devices by generating numerous sessions with unique IPs. The Scapy library is again used to create and dispatch these packets.

3) **Ransomware:** In this scenario, files on UE1's PC are encrypted by an attacker, who then demands payment for decryption. The attack encrypts a folder using the cryptography library [19] and follows and implements a Python-based ransomware attack outlined in [20].

4) **Buffer overflow:** This attack manipulates a program's execution flow to execute arbitrary code, potentially damaging the UE's OS. The method for this attack is based on instructions from [21].

C. 5G Dataset

The datasets collected from MEC1 and MEC2 using our 5G testbed have been combined for ease of result analysis and feature comparison. The dataset comprises 77 features, categorized as follows: 35 network features as defined by Hady et al. [22], 40 host features (detailed in Table I), and two label features (one for multi-label classification and the other for binary classification). Overall, the dataset includes 145k samples, consisting of 132k normal samples and 13k attack samples [23].

TABLE I. DATASET HOST FEATURES

Metric	Description
IMEI	International Mobile Equipment Identity
RTime	Packet sending time
Packet_num	Packet number
scputimes_user	Normal processes' CPU time
scputimes_nice	Prioritized processes' CPU time
scputimes_system	Kernel processes' CPU time
scputimes_idle	Idle CPU time
scputimes_iowait	CPU time waiting for I/O
scputimes_irq	Hardware interrupts CPU time
scputimes_softirq	Software interrupts CPU time
scputimes_steal	Virtualized CPU time
scputimes_guest	Normal virtual CPU time
scputimes_guest_nice	Prioritized virtual CPU time
scpustats_ctx_switches	Number of context switches
scpustats_interrupts	Number of interrupts
scpustats_soft_interrupts	Number of software interrupts
scpustats_syscalls	Number of system calls
svmem_total	Total physical memory size
svmem_available	Available memory size
svmem_percent	Percentage of used memory
svmem_used	Broadly consumed memory
svmem_free	Free memory size
svmem_active	Currently in-use memory

svmem_inactive	Currently unused memory
svmem_buffers	System metadata cache
svmem_cached	Other things cache
svmem_shared	Shared memory
svmem_slab	In-kernel data structures cache
ram_usage_warning	RAM usage warnings
sswap_total	Total swap memory in bytes
sswap_used	Used swap memory in bytes
sswap_free	Free swap memory in bytes
sswap_percent	Swap memory percentage usage
sswap_sin	Number of bytes swapped in from disk
sswap_sout	Number of bytes swapped out from disk
sdiskusage_total	Total disk size
sdiskusage_used	Used disk size
sdiskusage_free	Free disk size
sdiskusage_percent	Percent disk used
Boot_Time_with_date	Boot time and date
DTime	Packet delivery delta time

D. DRL model

The DRL model implements the DDPG algorithm, designed for continuous action spaces. This model was developed using the Keras library from the Tensorflow package [24]. It comprises two main components: the actor and the critic. The actor is tasked with selecting actions, while the critic evaluates these actions by estimating their value functions. Details of the actor and critic models, including their hyperparameters, are presented in Table II. The architecture includes three hidden layers: the first consists of 32 Gated Recurrent Unit (GRU) neurons, while the other two layers each have 48 fully connected neurons, following the setup outlined in [25]. An epsilon-greedy strategy is employed to adjust the action policies. Additionally, the learning rate dynamically changes based on the rewards received during the model's training phase.

TABLE II. DRL MODEL HYPERPARAMTER

Parameter	**Typical Value(s)**
Number of layers	4
Number of neurons per layer	32, 48, 48, 1
Number of episodes	10
Actor's initial learning rate	0.0001
Critic's initial learning rate	0.001
Discount factor	0.99
epsilon	0.1
Anomaly rate range	(-0.1, 0.0)
Optimizer	*Adam*
Activation function	*relu, tanh*

V. EXPERIMENTAL RESULTS

In this section, we discuss the analysis and evaluation of the dataset using the newly developed HDRL model. First, we explain the preprocessing steps applied to the dataset. Subsequently, we present the initial results obtained from the HDRL model, utilizing the collected dataset. This includes a comprehensive comparison between the usage of hybrid features, network-only features, and host-only features.

A. Dataset Preprocessing

Preprocessing the dataset is crucial for any ML method, including DRL models, to ensure accurate results. The preprocessing steps consist of dataset splitting, data encoding, normalization, and resampling, as follows:

1) **Dataset splitting:** The dataset is divided into three parts for training and evaluating the DRL model: 75% for training (10% of this portion is allocated for validation) and 25% for testing.

2) **Data encoding and normalization:** Categorical features are encoded, and the dataset is scaled using the label encoder and Standard Scaler from the scikit-learn package [26], respectively.

3) **Divide the features based on their types:** To compare hybrid, network-only, and host-only features, we have prepared three separate datasets. The first two consist of a single feature type, while the third contains both types. We then run the same model on these datasets and present their results in Section V.B.

B. Results

We compared the hybrid HDRL model to those using only network or host features. This comparison utilized two different metrics: the F_1 score and accuracy. The F_1 score, a harmonic mean of precision and recall, is a widely used metric in IDS results. Accuracy is defined as the ratio of correct predictions to total predictions. As these are initial results, The range for the anomaly rate, (-0.1, 0.0), was determined by taking the average action values from the training samples and adjusting it by subtracting 0.05 and then adding 0.05.

Fig. 4 shows the accuracy scores for all three types of features. As shown in the figure, the HDRL model achieved 58.17%. In contrast, the features from the net-only dataset achieved 91.76%, compared to a mere 24.99% for the host-only features. Despite the net-only model's high accuracy score, these results underscore that accuracy alone may not be the most reliable measure for IDS effectiveness. This is because the net-only IDS failed to identify nearly all attack

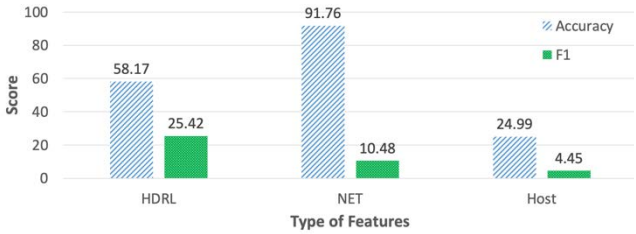


Fig. 4. Accuracy and F_1 scores comparison.

samples, rendering it essentially as effective as a model with a 0% detection rate, which still achieves a 91.42% accuracy score. However, combining both yields significantly better results than using either type individually.

Similarly, Fig. 4 presents the F_1 scores for all three types of features. Given that the F_1 score is more suitable for security applications, a clear improvement in IDS performance is observed when combining network and host features. The HDRL method's score was more than two times higher than the other two feature types. Nonetheless, based on predefined parameters from other research, these initial results may not fully showcase the HDRL model's potential. Future work would explore the impact of training episode numbers and the use of feature-reduction methods to decrease training time.

VI. CONCLUSION

With advancements in cybersecurity and next-generation infrastructures like 5G, there is a growing need for dynamic IDS that can adapt quickly to evolving attack patterns with minimal human intervention. This paper introduces a novel dataset for IoMT systems, created using an emulated 5G network that includes direct and indirect end users and MEC services. This dataset, a first of its kind, addresses the gap in 5G IDS datasets, facilitating the detection of zero-day attacks. We collected 145k samples, comprising normal samples and four types of attacks: MiTM, DDoS, ransomware, and buffer overflow. A hybrid DRL model for IDS, combining the strengths of Network IDS and Host IDS, was developed to evaluate this dataset. The analysis demonstrates the benefits of combining features compared to using only network or host features. Our results show that the HDRL model outperforms in terms of F_1 score and accuracy metrics by 25.42% and 58.17%, respectively. Future work will focus on increasing the training episodes for the HDRL model to optimize IDS performance further.

ACKNOWLEDGMENT

This work was supported in part by Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia.

REFERENCES

[1] Intel. "Understanding the Advantages of 5G." [Online]. Available: https://www.intel.com/content/www/us/en/wireless-network/5g-benefits-features.html [Accessed: Jan 25, 2024].

[2] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, "A Deep Reinforcement Learning Based Intrusion Detection System (DRL-IDS) for Securing Wireless Sensor Networks and Internet of Things," Cham, 2020: Springer International Publishing, in Wireless Internet, pp. 73-87.

[3] Y. F. Hsu and M. Matsuoka, "A Deep Reinforcement Learning Approach for Anomaly Network Intrusion Detection System," in *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, 9-11 Nov. 2020 2020, pp. 1-6, doi: 10.1109/CloudNet51028.2020.9335796.

[4] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated Reinforcement Learning: Techniques, Applications, and Open Challenges," *ArXiv*, vol. abs/2108.11887, 2021.

[5] H. Moudoud and S. Cherkaoui, "Empowering Security and Trust in 5G and Beyond: A Deep Reinforcement Learning Approach," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 2410-2420, 2023, doi: 10.1109/OJCOMS.2023.3313352.

[6] A. Virdis, G. Nardini, G. Stea, and D. Sabella, "End-to-end performance evaluation of MEC deployments in 5G scenarios," *Journal of Sensor and Actuator Networks*, vol. 9, no. 4, p. 57, 2020.

[7] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35-59.

[8] A. T. D. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110618, 03/01 2021, doi: 10.1016/j.rser.2020.110618.

[9] Q. Jiaju, Z. Qihao, L. Lei, and Z. Kan, "Federated reinforcement learning: techniques, applications, and open challenges," *Intelligence & Robotics*, vol. 1, no. 1, pp. 18-57, 2021, doi: 10.20517/ir.2021.02.

[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, 2019/07/17 2019, doi: 10.1186/s42400-019-0038-7.

[11] J. P. Farwell and R. Rohozinski, "Stuxnet and the Future of Cyber War," *Survival*, vol. 53, no. 1, pp. 23-40, 2011/02/01 2011, doi: 10.1080/00396338.2011.555586.

[12] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection," *Computers*, vol. 11, no. 3, p. 41, 2022. [Online]. Available: https://www.mdpi.com/2073-431X/11/3/41.

[13] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports*, vol. 12, no. 1, p. 15370, 2022/09/13 2022, doi: 10.1038/s41598-022-19366-3.

[14] NIC Mexico. "Jool." [Online]. Available: https://nicmx.github.io/Jool/en/index.html [Accessed: Jan 25, 2024].

[15] QOSIENT LLC "ARGUS + ML." [Online]. Available: https://openargus.org [Accessed: Jan 25, 2024].

[16] G. Rodola. "psutil." GitHub repository. [Online]. Available: https://github.com/giampaolo/psutil [Accessed: Jan 25, 2024].

[17] Scapy community. "Scapy." [Online]. Available: https://scapy.net [Accessed: Jan 25, 2024].

[18] J. Oreman. "python-netfilterqueue." GitHub repository. [Online]. Available: https://github.com/oremanj/python-netfilterqueue [Accessed: Jan 25, 2024].

[19] pyca. "Cryptography." [Online]. Available: https://cryptography.io/en/latest/ [Accessed: Jan 25, 2024].

[20] A. Fadheli. "How to Make a Ransomware in Python." [Online]. Available: https://thepythoncode.com/article/make-a-ransomware-in-python [Accessed: Jan 25, 2024].

[21] J. Erickson, *Hacking: the art of exploitation, 2nd edition*. No Starch Press, 2008.

[22] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," *IEEE Access*, vol. 8, pp. 106576-106584, 2020.

[23] *WUSTL-HDRL-2024 Dataset for Cybersecurity Research on Medical Applications in 5G Networks*. [Online]. Available: https://www.cse.wustl.edu/~jain/hdrl/index.html

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "{TensorFlow}: a system for {Large-Scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265-283.

[25] L. Zhang, Z. Pan, Y. Pan, S. Guo, Y. Liu, S. Xia, Q. Zheng, H. Li, and W. Bai, "A Hidden Attack Sequences Detection Method Based on Dynamic Reward Deep Deterministic Policy Gradient," *Security and Communication Networks*, vol. 2022, p. 1488344, 2022/01/28 2022, doi: 10.1155/2022/1488344.

[26] P. Fabian, "Scikit-learn: Machine learning in Python," *Journal of machine learning research 12*, p. 2825, 2011.