Journal of Information Systems Engineering and Management
2025, 10(17s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

DCITD: A Deep Q-Network Approach for Cyber Image Threats Detection

Israa Saad Mohammed[1], Dr. Yossra Hussain[2]
[2]*Supervisor*
University of information technology and communication/ Information Institute for Postgraduate Studies, Baghdad, Iraq, Ms202320744@iips.edu.iq
University of Technology/Computer Science Department, Baghdad, Iraq,yossra.h.ali@uotechnology.edu.iq

ARTICLE INFO

Received: 05 Dec 2024

Revised: 24 Jan 2025

Accepted: 08 Feb 2025

ABSTRACT

Cybersecurity threats Continuously develop and adapt. Posing serious risks for companies, governments, and individuals worldwide. Traditional methods for detecting these threats, which often rely on fixed rules and established patterns, are ineffective against attackers' dynamic and sophisticated tactics. Detecting cyber threats, especially for malware images, presents a considerable challenge for organizations and individuals. Conventional detection techniques, which often depend on fixed rules, are increasingly ineffective against the sophisticated strategies utilized by today's attackers. That calls for creating more sophisticated and intelligent cyber defense systems, integrating autonomous agents that can learn and make decisions without relying on human knowledge. This paper employed Reinforcement Learning techniques, which is one of the machine learning fields based on trial and error for learning, to propose the Detection of Cyber Image Threats by the DQN (DCITD) model for malware detection system leveraging Deep Q-Networks (DQN) integrated with image-based reinforcement learning. The model uses a Convolutional Neural Network (CNN) to feature extraction and incorporates multithreading to optimize experience replay during training. The DCITD model, utilizing Deep Q-Network (DQN) architecture, showcases a permanent auto-learning feature within a network setting, allowing for detecting various network threats through an automated trial-and-error process while steadily refining its detection capabilities. The paper is based on thorough experimentation utilizing the Blended malware dataset, and the results reveal that the proposed DCITD model excels in recognizing a wide array of threats and outperforms similar machine-learning techniques. Those techniques produce a fusion of features to build a model that can be used to recognize and classify the malware images into 31 malware families, then evaluate the performance of malware classification by identifying unique malware families and tracking detection accuracy that reached 98%.

Keywords: Cybersecurity, Reinforcement learning, Deep learning, DQN, Image threats, Malware

I. Introduction

The fast growth of digital technologies has transformed industries. Furthermore, it significantly increased organizations' vulnerability to cybersecurity threats. These threats, spanning from phishing, ransomware, and Malware to Distributed Denial-of-Service (DDoS) attacks, can cause severe financial damage and irreparable harm to reputations. The challenge lies in swiftly detecting these threats—especially the more complex ones represented as images—while minimizing detection time, reducing false positives, and adapting to the evolving landscape of attack vectors. Traditional detection systems, which predominantly rely on signature-based or heuristic methods, are often effective against known attack patterns. However, they frequently fail when confronted with zero-day threats and sophisticated attacks disguised as legitimate activities. This shortfall has created an urgent need for intelligent and adaptive systems capable of learning and addressing the dynamic nature of modern cyber threats. The DCITD model, based on Reinforcement Learning (RL), emerges as a promising solution to address this. Unlike conventional supervised learning frameworks, RL emphasizes training agents to interact with their environment and determine optimal strategies through trial and error. The DCITD model introduces an innovative combination of network intrusion detection techniques by employing a Q-learning-based RL framework integrated with a deep feed-forward

neural network. This approach enhances network threat identification, offering more effective self-learning capabilities. In the field of RL methods, Deep Q-networks (DQNs) have garnered substantial attention for their ability to combine the representational power of deep learning with RL decision-making proficiency. This makes them especially well-suited for addressing high-dimensional, complex, and sensitive datasets, such as those involving image-based threats. This paper explores using DQNs for accurate and reliable cybersecurity threat detection. The key contributions of our proposed method are as follows:

- Detection of threats in cybersecurity images using Deep Reinforcement Learning technologies
- Recognized and classified multi-malware families using DQN
- reduced the complexity and time-consumption using multi-threads
- CNN is effectively used for feature extraction and feature selection, respectively.
- Evaluation of this model is depicted using various performance measures such as accuracy, precision, recall, f1-score, and training loss.

The other parts of the paper are prepared as follows: Section 2 discusses related work, while Section 3 outlines the details of the proposed methodology. The 4th section presents the results and discusses the experimental results and their importance. Section 5 concludes the paper.

II. Related works

Many threat classification and identification research in images or text works have been investigated based on machine learning approaches, including image-based feature extraction, hybrid models, and deep learning approaches. In [1] (Thakur, 2020), The authors proposed a method for classifying Android Malware by transforming it into image sections and using the GIST algorithm to extract features from the DREBIN dataset. These features are classified utilizing machine learning algorithms like SVM, KNN, RF, and NB. The SVM classifier achieved an accuracy of 92.7% with Android manifest image files. The study concludes that visualization techniques combined with machine learning can improve malware classification accuracy. In [2](O Aslan et al., 2021), The authors proposed a malware classification framework to detect and classify malware variants from the Malimg dataset and Malevis dataset, achieving 96.5% accuracy, Microsoft BIG 2015 dataset achieved 97.48% accuracy using the ResNet-50 and AlexNet methods to achieve 97.78% accuracy. However, in [3](Ullah et al., 2022), The researchers presented a malware detection system for Android apps using a hybrid transfer learning approach and multi-model image representation. It combines textual and texture features of network traffic to detect Malware. The proposed method is tested on two datasets, CIC-AAGM2017 and CIC Mal Droid 2020, and has achieved 99% accuracy in malware classification and detection. The system leverages word2vec embeddings and convolutional neural networks to feature extraction and classification. In [4](Golubev et al., 2022), the authors proposed intrusion detection in cyber-physical systems by transforming raw network packets into grayscale images. The creators analyze existing methods and introduce a technique that leverages convolutional neural networks (CNNs) for detecting network attacks. Resulted in 99% accuracy using the SWaT dataset. The authors (N. Gyamfi et al., 2022) in [4]The paper introduces a malware detection framework named D-WARE. that utilizes deep learning techniques to identify malicious software, using (PCA) for feature extraction Principal Component Analysis and Particle Swarm Optimization (PSO) for feature selection and classification via Convolutional Neural Networks (CNN). Applying D-WARE on the MalImg dataset resulted in 96% accuracy. The authors in [5] (Ren et al.,2023) explored the utilization of deep reinforcement learning (DRL) for cyberattack detection. It proposes a framework utilizing an agent-based model that continuously learns and adapts within a dynamic network security environment—using the double deep Q-network (DDQN) and policy gradient (PG) models in improving cyberattack detection outcomes across three datasets: NSL-KDD, CIC-IDS-2018, and AWID and resulting in 96.8% accuracy and 96.3% F1 score for CSE-CIC-IDS2018 and 99.1% for NSL-KD. Meanwhile, the authors (S. Mohanty, S. Nanda, R. Rout, et al.) are in [8]. The authors proposed a supervised logistic regression using machine learning classifiers, Naive Bayes, k-Nearest Neighbor, and Decision Tree to identify cybersecurity attacks. They created a dataset with 29 lexical features of URLs and trained the model using these features. The experimental results showed that the proposed machine learning model achieved a 94.1% accuracy in detecting different attacks based solely on URLs' lexical or structural features—table 1. Abstract Various studies have explored techniques for malware and cyberattack detection. Methods.

no	Ref & Year	The Aim	Methods used	Dataset	Classifier Result
	[1], 2020	Classify Android malware through the utilization of visualization techniques.	Naive Bayes	The DREBIN dataset	76.81% accuracy
			Random Forests (RF)		91.01% accuracy
			K-Nearest Neighbors (KNN		92.38% accuracy
			Support Vector Machines (SVM)		92.7% accuracy
	[2], 2021	malware classification framework to detect and classify malware variants	ResNet-50 and AlexNet	Malimg dataset	97.78% accuracy
				Malevis dataset	96.5% accuracy
				Microsoft BIG 2015 dataset	94.88% accuracy
	[3] , 2022	present a malware detection system based on transfer learning and multi-model image representation	Convolutional Neural Networks (CNNs)	CIC-AAGM2017 and CIC Mal Droid 2020 datasets	99% accuracy
	[5], 2022	malware detection utilizing deep learning techniques for identifying malicious software	Convolutional Neural Networks (CNNs), PSO, PCA	MalImg dataset	96% accuracy
	[4], 2022	intrusion detection in cyber-physical systems by transforming raw network packets into grayscale images.	Pre-trained models (ResNet34, MobileNetV3-small) in conjunction with a bespoke CNN	SWaT dataset	99% accuracy
	[6], 2023	cyberattack detection within a dynamic network security environment	the double deep Q-network (DDQN) and policy gradient (PG) models	NSL-KDD	99.1% accuracy 99% F1scor
				CIC-IDS-2018	96.8% accuracy 96.3% F1scor
	[8] ,2023	detection of various cybersecurity attacks	Logistic Regression, Naive Bayes, k-Nearest Neighbor, Decision Tree,	dataset with 29 lexical features of URLs and trained the model using these features	94.1% accuracy

Table 1. Summary of related works

III. Methodology

The methodology phase is based on several steps to enhance detection and classification accuracy and response efficiency against sophisticated cyber threats. The proposed DCITD model focuses on developing a cyber malware image threat detection and classification system using the Deep Q-Network (DQN) algorithm. DQN, one of the Reinforcement learning models, enables the system to learn optimal detection strategies by interacting with dynamic network environments, Using cyber threats in image representations. The model leverages convolutional neural networks (CNNs) for feature extraction and decision-making. This approach allows adaptive learning and real-time detection of complex and evolving cyber threats.

Malware is malicious software that executes malicious actions on the victim's machine to endorse scams. When the users click this infected URL, the Malware like ransomware, virus, trojan, or any other type of Malware downloaded automatically will compromise both the machine and network. Malware is used to steal sensitive data and payment information, disrupt operations, and demand payment [7], as shown in Figure 1. The diagram illustrates a malware attack when a malicious email tricks the victim by downloading a RAR file containing a Dropper EXE, which decrypts and injects the Bandook RAT into iexplore.exe. The infected system then connects to the attacker's C&C server, allowing remote control and the download of additional malicious DLL extensions. This process makes full system compromise and unauthorized access.

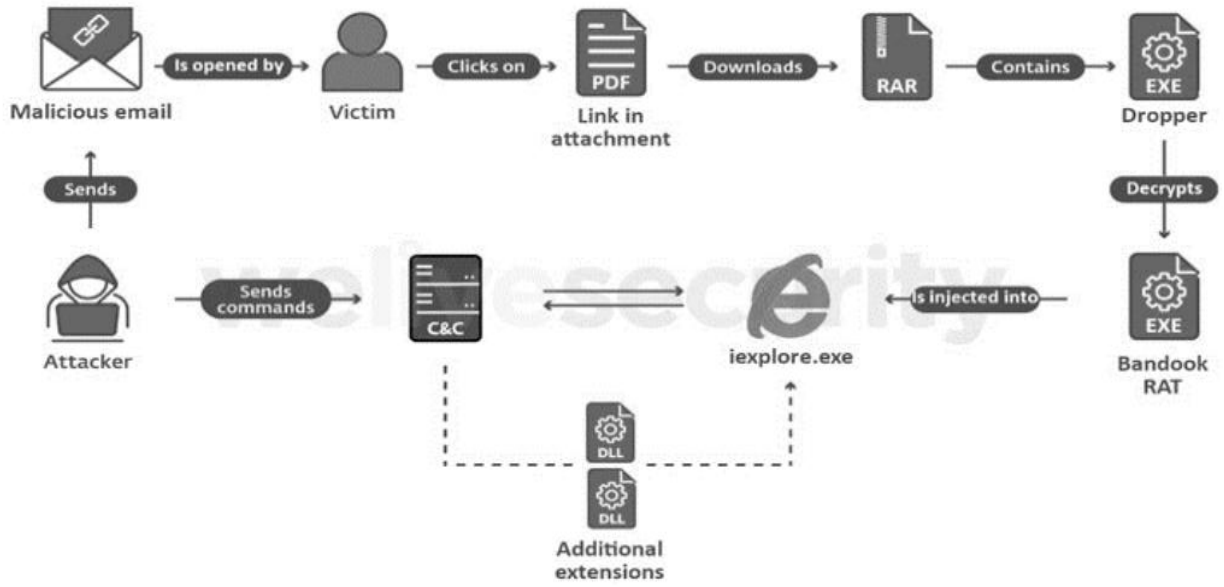


Figure 1. explains the malware attack life cycle.[6]

Datasets description

The two widely recognized datasets utilized for the malware image classification category are the MalImg and the Malevis datasets. The MalImg dataset exhibits a significant imbalance regarding class distribution, whereas the Malevis dataset is characterized by its equitable class distribution. Both datasets have been amalgamated to construct a unified dataset incorporating all classes from the Malevis dataset and five classes from the MalImg dataset [7]. The blended malware dataset encompasses 13,700 images with RGB and Grayscale byteplot images, resulting from integrating the two datasets into one dataset. Ishwarya Suresh curated it on Kaggle; this dataset is a comprehensive resource specifically engineered for malware detection tasks. It comprises both malicious and benign files, structured into 31 classes of malware family to facilitate researchers and data scientists in deploying machine learning models for cybersecurity applications [8]. The dataset's classes, family types, and image count for each are described in Table 2. The dataset is organized into two directories, one designated for training and the other for validation sets. Each of these sets contains samples from 31 distinct classes, which can be utilized for training and assessing models to identify Malware based on patterns present within the data. Figure1. Illustrate the dataset samples, wherein the first row contains two image samples of the Blended malware dataset while the second row describes the general structure of malware images.

Table 2. The malware classes with the number of images in each class.

no	Class (family type)	Number of Images
1.	Adposhel	494
2.	Agent	470
3.	Allaple	478
4.	Alueron.gen!J	198
5.	Amonetize	497
6.	Androm	500
7.	Autorun	496
8.	BrowseFox	493
9.	C2LOP.gen!g	200
10.	Dialplatform.B	177
11.	Dinwod	499
12.	Elex	500
13.	Expiro	501
14.	Fakerean	381
15.	Fasong	500
16.	HackKMS	499
17.	Hlux	500
18.	Injector	495
19.	InstallCore	500
20.	Lolyda.AA1	213
21.	Lolyda.AA2	184
22.	MultiPlug	499
23.	Neoreklami	500
24.	Neshta	497
25.	Regrun	485
26.	Sality	499
27.	Snarasite	500
28.	Stantinko	500
29.	VBA	500
30.	VBKrypt	496
31.	Vilsel	496

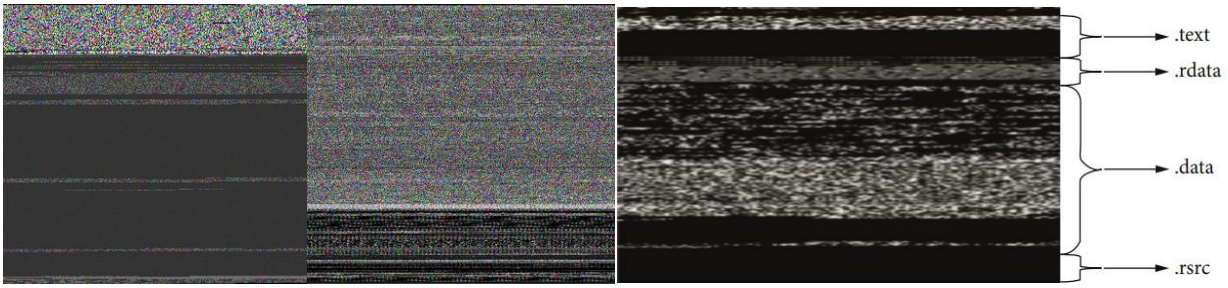


Figure 2. Sample of the malware image dataset.

Reinforcement learning (RL) is a machine-learning field, like supervised or unsupervised learning. It learns the best actions an agent needs to perform to maximize its rewards in a particular environment, sometimes called seem supervised. In RL, the agents learn by interacting with an environment. The agent monitors the state of the environment, performs actions, and receives rewards or penalties based on the results of those actions. Over time, the agent learns the optimal policy to maximize cumulative rewards. Reinforcement learning components involve agents, states(S), and actions per state (A). Agents evolve from the state when they act to learn how to react; finally, agents make decisions; the Reinforcement learning essential component is shown in Figure 3.

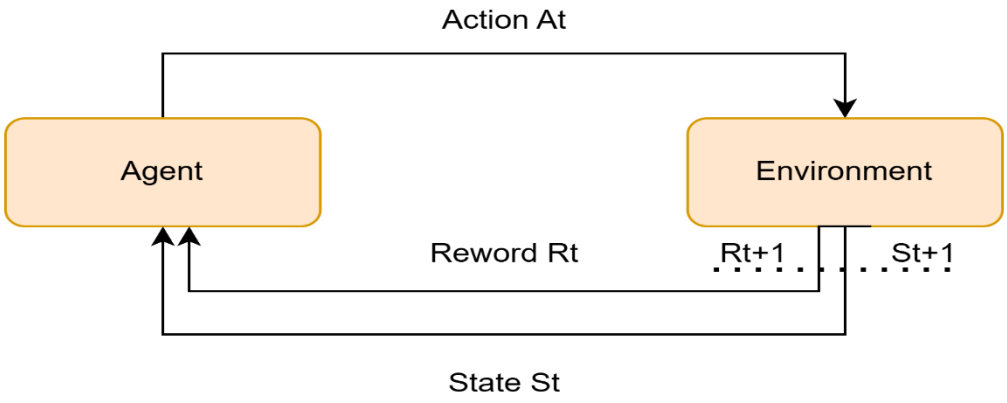


Figure 3. Reinforcement learning (RL) components

Reinforcement learning (RL) can be divided into two main categories: model-based RL and model-free RL. Model-based reinforcement learning algorithms are approaches that attempt to explain the Environment and create a model simulated to it like world models,12 A, MBVE, and Alpha zero algorithms, while Model-free reinforcement learning algorithms are approaches that will also have two type categories: policy-based only update its policy by interacting with it is Environment and observing the rewards like policy gradient, PPO, TRPO. Value-based reinforcement learning algorithms like SARSA, Q-learning, DQN, DDQN, and Distributional Q-Learning are approaches where the agent learns a value function that estimates how good it is to be in a particular state (or to perform a specific action in a state). This paper will focus on DQN, one of the value-based Reinforcement learning algorithms.

DQN The Deep Q-Network is one of the model-free reinforcement learning algorithms introduced to address various complication problems within computer vision. It integrated the rules of the classical Q-learning methodology with deep convolutional neural networks (CNNs). A primary impetus for the development of DQN was to address the limitations and capacity associated with the Q-table utilized in Q-learning, which can adapt only a finite quantity of states. In contrast, real-world applications may necessitate the management of a substantial or potentially infinite array of states. DQN incorporated an experience replay mechanism, facilitating the random sampling of a small subset of tuples from the replay buffer throughout the training phase. This approach substantially mitigated the correlations among the sampled data, thereby enhancing the overall robustness of the algorithm. DQN employed a deep convolutional neural network to encapsulate the current Q function and utilized a separate network to establish the target Q value. Implementing the target Q value network effectively diminished the correlation between the current and target Q values.[9]

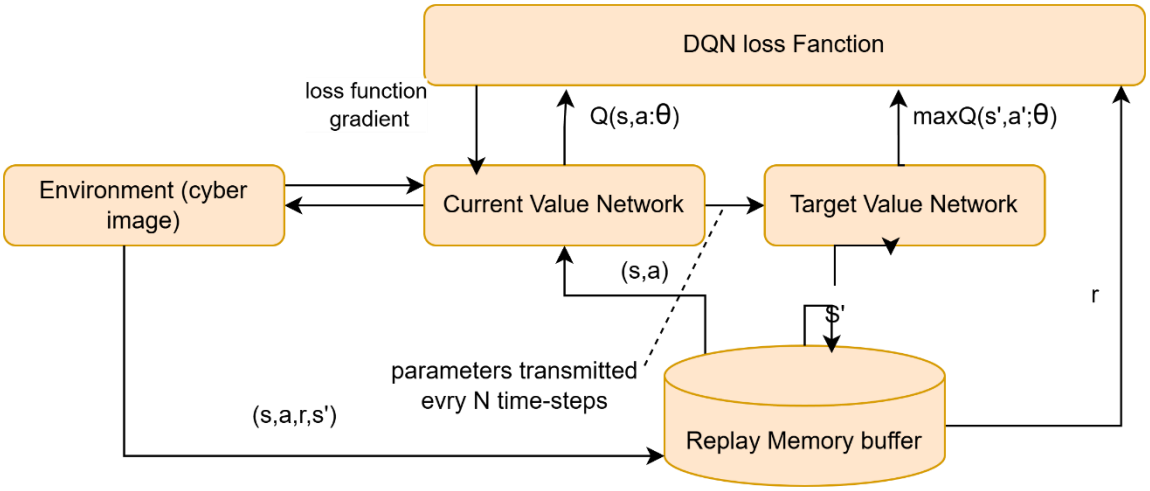


Figure 4 The operational workflow of the Deep Q-Network (DQN)

Figure 4 illustrates the workflow of the Deep Q-Network DQN; this schematic delineates the mechanism by which a Deep Q-Network functions, elucidating the progression of information within a reinforcement learning framework. Environment represents the agent's domain, encompassing cybersecurity imagery that embodies potential hazards. The Environment delineates the current state (the existing conditions) and provides rewards (feedback) contingent upon the agent's actions. The Current Value Network constitutes the principal neural network that forecasts the Q-value corresponding to a specific state-action pair (Q(s, a; θ)). The Q-value encapsulates the efficacy of executing a particular action within a defined state. This network's parameters (θ) are subject to modification throughout the training process to enhance predictive accuracy. The Target Value Network is a supplementary neural network that supplies a consistent benchmark from which the current value network can derive insights. It computes the maximum prospective Q-value (max Q(s', a')) for the subsequent state, which is then utilized to amend the current value network. The parameters undergo updates at less frequent intervals (every N time step) to maintain stability during the learning process. The Replay Memory Buffer archives the agent's historical experiences as tuples: (state, action, reward, next state) or (s, a, r, s'). Rather than instantaneously assimilating each experience, the agent randomly sampled past experiences during training, thereby mitigating data correlations and enhancing learning efficacy. The DQN Loss Function inspects the inconsistency between the suggested Q-value (taken from the present value network) and the aimed Q-value (collected from the target value network). The gradient of this loss function is employed to recalibrate the parameters (θ) of the current value network, thereby facilitating improved predictions over time.

Proposed model

The Dynamic and Adaptive Learning paradigm effectively addresses cybersecurity challenges stemming from evolving threats like malware images and zero-day attacks, which often display changing temporal patterns. This paper introduces the DCITD model for detecting and classifying cyber image threats via Deep Q-Networks, which is designed for Cybersecurity and image-based threat detection. The model goes through several steps, as shown in Figure 5. The integration of Deep Q-Networks (DQNs) combined with multithreading programming libraries to leverage reinforcement learning (RL) for adaptive and intelligent threat identification in complex systems. By interacting with their Environment, DQNs continuously improve their performance, enabling them to adapt to dynamic threat landscapes without frequent retraining. This makes them highly effective in detecting new and sophisticated image-based threats.

Image-based threats like malicious image payloads involve complex and nonlinear data patterns. DQNs handle high-dimensional inputs like images using advanced neural architectures, particularly Convolutional Neural Networks (CNNs), to extract subtle visual features and anomalies. This capability is crucial for recognizing hidden malicious elements in images. DCITD model consists of a Current Q-Network and a Target Q-Network, as shown in Figure 5, which work together through gradient updates and replay memory to improve learning stability. The Replay Memory Buffer stores past experiences to enhance learning by sampling and replaying them. The system continuously refines its predictions by minimizing the DQN loss, ultimately leading to accurate threat detection. This approach enables adaptive and efficient detection of evolving cyber threats. Moreover, image threat detection involves complex

decision-making, where systems must determine whether to investigate specific image regions further or classify them as malicious. DQNs excel in this sequential decision-making process, developing optimal strategies to detect threats while minimizing false positives and resource usage.

A significant advantage of DQNs is their automatic learning ability, which allows them to extract high-level features from raw image data without relying on manually crafted features, which are often ineffective against zero-day threats. Additionally, DQNs are well-suited for real-time threat detection, responding swiftly to potential risks—an essential capability in Cybersecurity. Their resilience against adversarial attacks is another strength, as they are trained across diverse attack scenarios, enabling them to identify and adapt to deceptive tactics, particularly in image-based threats where attackers may obscure malicious intent.

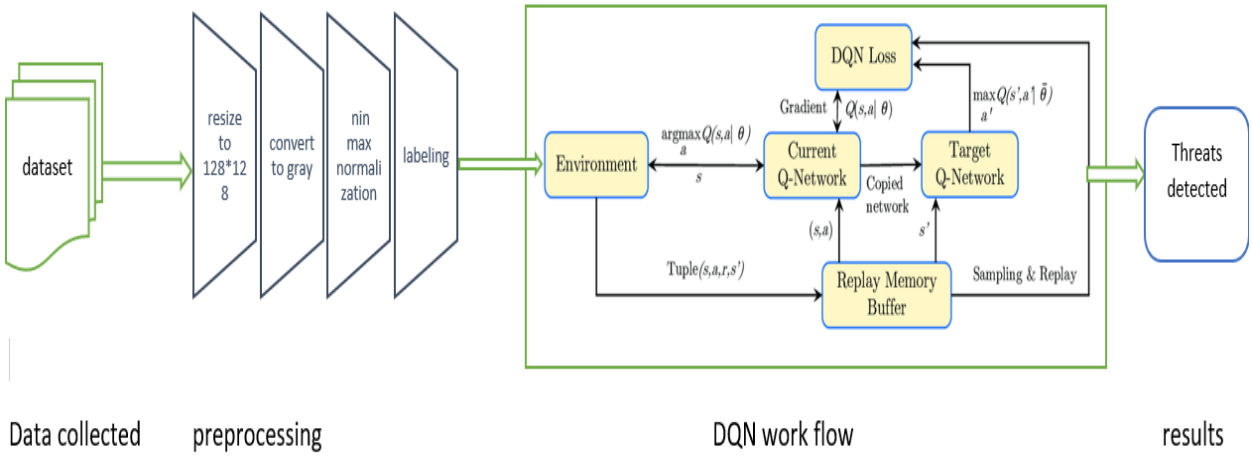


Figure 5.DCITD model Deep Q-Network for Cyber Threats Detection

Data collected: All samples are collected from blended malware image classification datasets. One will run with the proposed model separately to ensure the improvement of the DCITD model in detecting different types of threats in image-based Malware. After loading, the dataset is decoded into a usable tensor format for TensorFlow. Loads the image as an RGB image; three channels, regardless of the original format, to ensure consistent input data format; the dataset is loaded at 64 batch size each Episode, and the DQN expects a consistent state space, so all images must be adequately loaded.

Table 3. Dataset split partitions for training and testing

Total images	Number of images for training	Number of images for testing
13,700	10,997	2750

The preprocessing step in constructing the model is preprocessing the dataset. The first step in preprocessing converts the image from RGB (3 channels) to Grayscale (1 channel) to reduce input dimensionality, making training faster and less memory-intensive, mainly since Malware detection usually relies on patterns and structures rather than colors—Simplifies feature extraction by focusing on texture and structure. The second step in preprocessing is to resize images into 128x128 pixels to ensure a uniform input size for the convolutional neural network (CNN) used in the DQN. However, the smaller, fixed-size images reduce computational load and speed up training. Moreover, it avoids inconsistencies caused by varying image sizes. The third step is normalization by Scales pixel values from [0, 255] to [0, 1]. To stabilize training by keeping input values within a small, consistent range that prevents large input

values from causing unstable gradients during backpropagation and helps the neural network converge faster. Yielding NumPy arrays and ultimately labeling by invoking the aforementioned functions. Figure 6. illustrates samples of the post-preprocessing steps. Then designating labels (1 for "malware" and 0 for "benign"). After these preprocessing steps, the DQN agent receives clean, uniform, and optimized input data to improve efficient training and accurate detection. Finally, the feature dataset is loaded in parallel, processing images concurrently through multiple threads for enhanced efficiency. These features become the foundational input for the learning algorithm. After preprocessing, the dataset is divided into training and testing sets in an 80-20 ratio. This division ensures that the model can generalize effectively to previously unseen data. The results of this dataset splitting are presented in Table 3. The training set is used for the model's learning phase, while the testing set is designated for evaluating the performance of the final model.

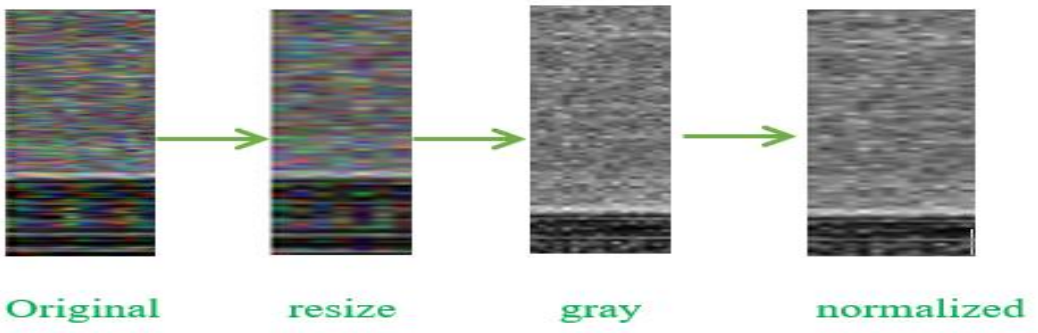


Figure 6 samples the preprocessing stages of the DCITD model.

Deep-Q learning wherein the agent initializes the environmental state by sampling a traffic sample and striking a balance between exploration (random actions) and exploitation (maximization of Q-values). The agent acts, observes the subsequent state and reward, and archives the experience in a replay buffer for training purposes by trial and error as Mini-batches (64) extracted from the buffer are employed to compute target Q-values and to customize the Environment. The subsequent phase in the DCITD model involved the delineation of the Environment that was created using the Gym library, where the agent interacts with the dataset by classifying each image and receiving rewards for correct predictions based on a reward system and representing each traffic sample as a state (s) and specifying two potential actions (a). Specifically, a is designated as zero to denote classification as benign, whereas a is assigned a value of one to signify classification as malicious.

Furthermore, rewards were defined by (r), where the correct classification yields reword+1 and the incorrect classification yields reword-2. Thereby providing feedback through the reward system. The Episode ends when all images have been processed in the same manner. The following procedural step encompassed a training loop over (150) episodes. The target network undergoes periodic updates by updating the Q-network via gradient descent methodologies, facilitating the agent's progressive enhancement in decision-making capabilities over time through continuous learning derived from its Environment and prior experiences. The amalgamation of neural networks and reinforcement learning renders the model particularly efficacious for intricate tasks such as image threat detection within Cybersecurity.

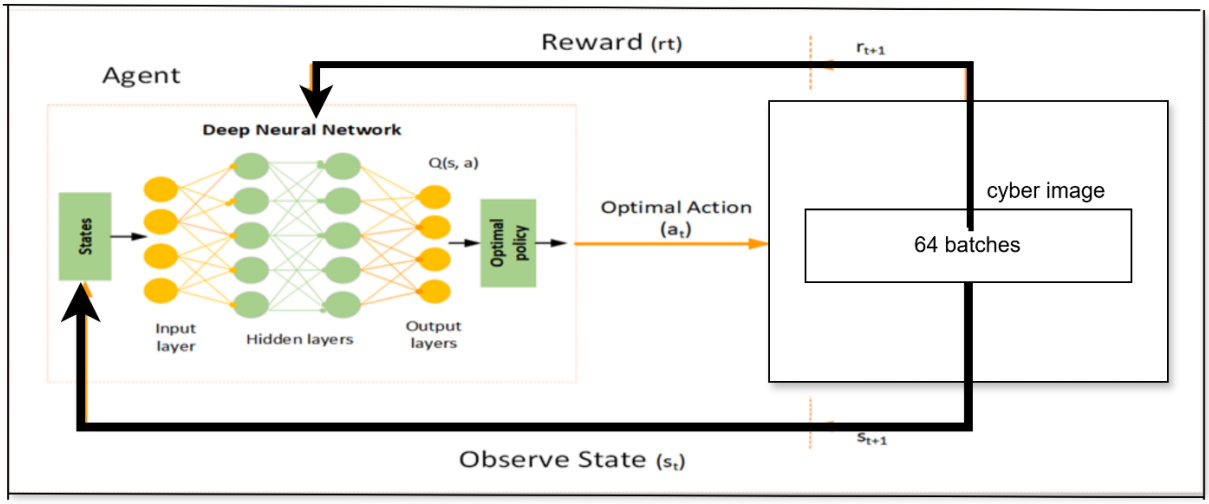


Figure 7. Deep neural network agent observes in DQN

The proposed DCITD model architecture is structured with an input layer that processes normalized network traffic features. It encompasses two hidden layers characterized by ReLU activation functions, which serve to identify and extract salient features. The output layer is responsible for calculating Q-values to ensure whether the traffic is categorized as usual or malicious, employing the Bellman equation for the iterative refinement of these Q-values. As shown in Figure 7, the Environment's state selects optimal actions based on Q-values, receives rewards, and updates its policy iteratively. Additionally, the model integrates a reward system to enhance efficiency, and it utilizes experience replay to learn from archived data and a target network to uphold stability through periodic weight updates. Table 4 depicts the outcomes of neural network parameters after training the DCITD model. The total parameters of the Neural Network after training were 22,176,008 (84.59 MB) trainable Parameters:7,392,002 (28.20 MB) and Non-Trainable Parameters 0 (0.00 B), Optimizer Parameters14,784,006 (56.40 MB).

Table 4. Neural network parameters after training DCITD proposed model

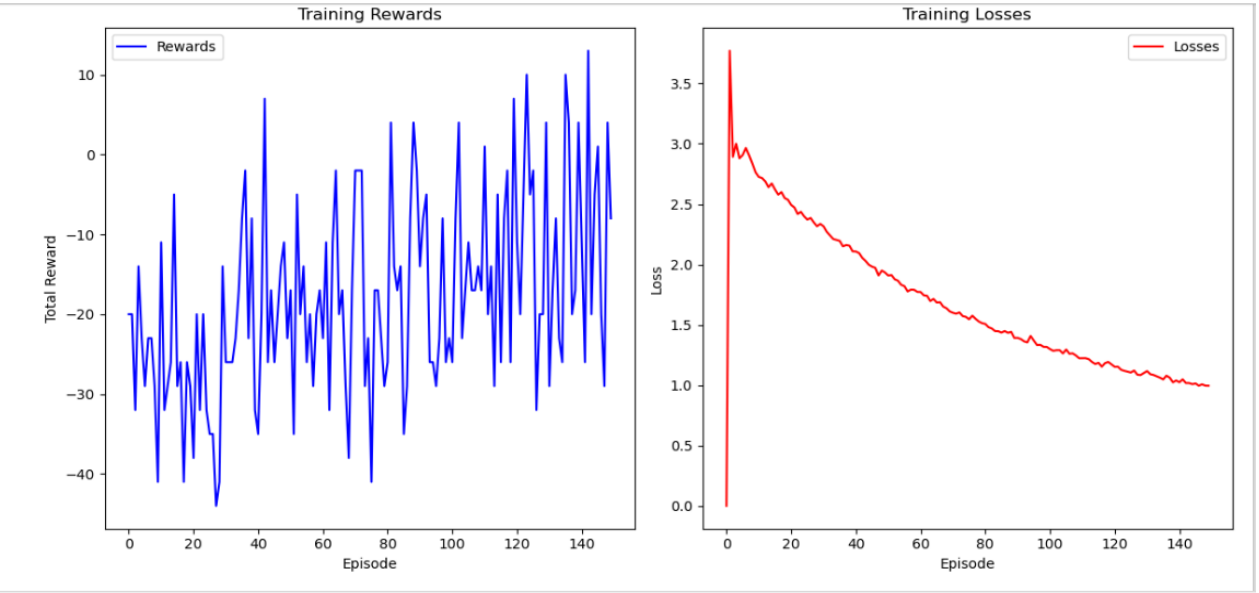
Layer (type)	Output Shape	Param #
conv2d_60 (Conv2D)	(None, 126, 126, 32)	320
max_pooling2d_60 (MaxPooling2D)	(None, 63, 63, 32)	0
dropout_90 (Dropout)	(None, 63, 63, 32)	0
conv2d_61 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_61 (MaxPooling2D)	(None, 30, 30, 64)	0
dropout_91 (Dropout)	(None, 30, 30, 64)	0
flatten_30 (Flatten)	(None, 57600)	0
dense_60 (Dense)	(None, 128)	7,372,928
dropout_92 (Dropout)	(None, 128)	0
dense_61 (Dense)	(None, 2)	258

Key components of the DCITD Model Architecture include the State-Value Stream, which estimates the value associated with a given state, and the Advantage Stream, which assesses the advantage relative to each action. The Replay Buffer stores past experiences (state, action, reward, next state, done) for training purposes. The model

explores random actions based on probability (epsilon) to maintain equilibrium between exploration and exploitation. By utilizing the trained model, please enable it to predict the optimal action (classifying Malware or benign). The training and evaluation of the DCITD model involved training with 80% of the dataset, utilizing threading to enhance time and resource consumption. This training was divided into segments (episodes) and parallelized using four threads utilizing the (ThreadPoolExecutor) library in Python to expedite the training process. Afterward, the factor was evaluated by applying it to the remaining 20% of the dataset, during which different evaluation metrics, precision, precision, recall, F1-Score, and confusion matrix, were calculated.

IV. Results and Discussion

This paper studies the effectiveness of Deep Networks (DQN) in identifying and classifying different types of cybersecurity image threats. By leveraging the power of reinforcement learning, the proposed DCITD model demonstrates a dynamic learning capability that adapts to evolving image attack patterns. The proposed model was trained and evaluated on the Blended Malware dataset to successfully detect and classify various malware families with high accuracy and precision. The DCITD model accurately detected and classified 6,103 malware threats across diverse families during training and testing, improving the strength model performance in distinguishing malicious files from benign ones. The model accomplished an accuracy of 98%, precision of 99 %, recall of 98%, and an F1-score of 99%. However, it reached just 0.02 of the error rate; Figure 8. illustrates the outcomes derived from the implementation of the Deep Convolutional Inverse Transfer Learning (DCITD) methodology on a malware dataset, wherein the graphical representations delineate the reward metrics, training loss, and confusion matrix about false and true alarms across a testing phase where the blue squire refer to the accurate optimistic prediction of Malware detected family that authentic identify 63 samples from total 64 where the third blue light squire refer to accurate pessimistic prediction of only one sample. The model's results demonstrated that it is highly effective in identifying and classifying various malware image threats. However, The minimum training loss was 0.9952, indicating room for optimization in the training process. The frequency distribution of detected Malware highlights the model's ability to generalize across different malware types, effectively learning complex patterns associated with malware behavior among the most frequently detected families, from the most frequently Androm was identified with 505 samples to the lowest-frequency detections included C2lop.gen!g (80 samples), that underscoring of the system's capability to identify rare or obscure as soon as more complex behaviors, that posing a more significant challenge for detection threats The Implications of The DCIT model for Cybersecurity was ability to detect a wide range of malware families confirms and improved reinforcement learning's powerful and proactive tool in cybersecurity defenses.



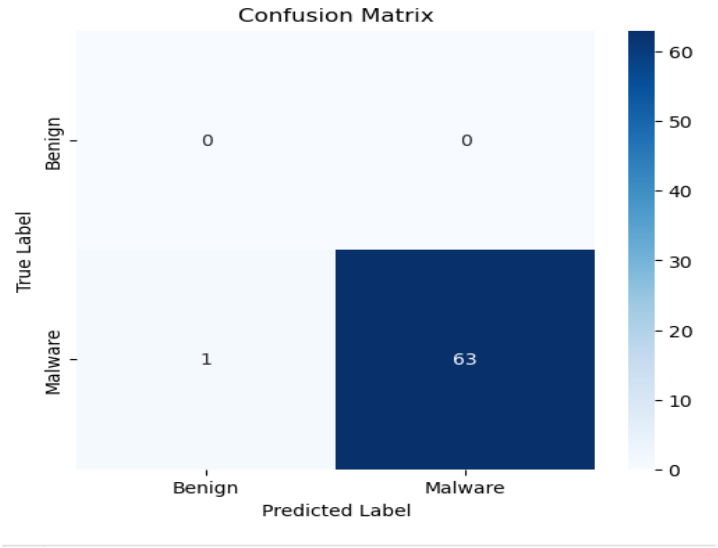


Figure 8. Results Obtained from the Learning (DCITD) Methodology Applied to a Malware Dataset

The DQN model can enhance malware detection systems by continuously learning and adapting to new threats. The results of the detected and classified malware families and their frequencies are summarized in the training and test phases distribution bar plot in Figure 9. The results highlight the model's effectiveness in generalizing across diverse threats. The frequency distribution underscores the model's capability to detect common and rare malware families, achieving a high detection rate and showcasing its potential for real-world cybersecurity applications. This work demonstrates the significant potential of deep Reinforcement learning in combating cybersecurity threats.

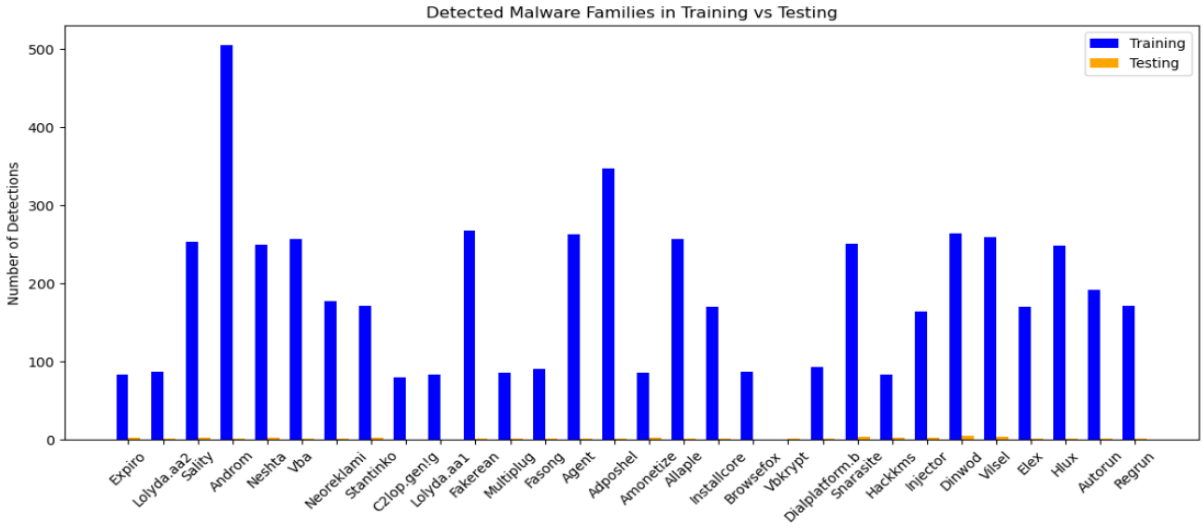


Figure 9. Types and Malware families frequently detected by the proposed DCITD model

When comparing the DCITD model with the [4], it introduces a malware image detection framework designated as D-WARE. Implementing the MalImg dataset resulted in an accuracy of 96% but has several significant limitations, such as dependencies on data quality, the dynamic nature of Malware, overfitting risks, and computation complexity. Those who made the model perform poorly in this issue and cannot detect new or unknown Malware. These issues are solved by the DCITD model whenever different ranges and frequencies of malware families are detected and classified. While the dynamic nature of Reinforcement learning DQN algorithm based on continuous learning enables the DCITD model to handle the newest Malware, complexity computation is solved using a multi-threaded and batch system. However, when comparing the DCITD model with the[2] hybrid deep-learning-based architecture(HDLA) model to detect and classify malware families from the Malimg dataset that can classify 25 families and achieved 97.78% accuracy, and the Malevis dataset achieved 96.5 % accuracy. The Microsoft BIG 2015 dataset achieved 94.88% accuracy. At the same time, the DCITD model achieved 98% accuracy. The divergent performance outcomes between

the two models are delineated in Table 5. In a broader context, the proposed DCITD model distinguishes itself as a state-of-the-art solution, achieving superior performance and providing exceptional capabilities in detecting threats within network data flows represented as images analogous to the datasets of the referenced model.

Table 5. Comparison performance between DCITD and D-WARE, HDLA models:

Model	Dataset	Number of images	Numbers of malware\s family detected	Accuracy	F1-score	Precession	Recall
D-WARE	MalImg	9458 images	8 family	96%	98%	96%	91%
HDLA	MalImg	9339 images	25 family	97.78%	97.79%	97.80%	97.78%
	Malevis	9100 images	25 family	96.5%	94.5%	97.1%	94.9%
	Microsoft BIG 2015	20000images	9 family	94.88%	89.88%	92.47%	91.31%
Proposal work DCITD	MalImg + Malevis	13700 images	29 family	98%	99%	99%	98%

V. Conclusion

In conclusion, this paper demonstrates the significant potential of Deep Networks (DQN) for cybersecurity applications, particularly in detecting and classifying malware threats from image-based data. By leveraging reinforcement learning, the proposed model showcased its ability to dynamically learn and adapt to evolving malware patterns, achieving a high accuracy of 98%. The model successfully identified and classified various malware families, including commonly occurring threats like Androm and Adposhel and rare families like C2lop.gen!g and Hackkms malware types, highlighting its robustness and versatility. The frequency distribution of detected malware families further underscores the system's capability to generalize across different threat types while maintaining high detection rates. With its ability to accurately classify over 6,103 malware threats and detect frequent and rare instances, the DCITD model is an effective tool for modern cybersecurity challenges. These results highlight the transformative potential of deep Reinforcement learning in combating cyberattacks and emphasize the importance of further research to improve scalability and address real-world dataset challenges. Overall, the paper reinforces the value of AI-driven solutions in safeguarding digital systems against ever-evolving cybersecurity threats.

For future improvements, even if the proposed DCITD model shows promising results, additional feature extraction techniques or training with a more diverse dataset can enhance detection accuracy, particularly for less common malware families. Furthermore, the application can be used to check, detect, and classify the malware threats included in images, yielding superior performance outcomes and enabling the development of a threat detection application characterized by ease of use and clear visual representation.

Reference :

[1] D. Thakur, “Classification of Android Malware using its Image Sections,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 6151–6155, 2020, doi: 10.30534/ijatcse/2020/288942020.

[2] O. Aslan and A. A. Yilmaz, “A New Malware Classification Framework Based on Deep Learning Algorithms,” *IEEE Access*, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[3] F. Ullah, S. Ullah, M. R. Naeem, L. Mostarda, S. Rho, and X. Cheng, "Cyber-Threat Detection System Using a Hybrid Approach of Transfer Learning and Multi-Model Image Representation," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155883.

[4] S. Golubev, E. Novikova, and E. Fedorchenko, "Image-Based Approach to Intrusion Detection in Cyber-Physical Objects," *Inf.*, vol. 13, no. 12, 2022, doi: 10.3390/info13120553.

[5] N. K. Gyamfi, N. Goranin, D. Ceponis, and A. Čenys, "Malware Detection Using Convolutional Neural Network, A Deep Learning Framework: Comparative Analysis," *J. Internet Serv. Inf. Secure.*, vol. 12, no. 4, pp. 102–115, 2022, doi: 10.58346/JISIS.2022.I4.007.

[6] K. Ren, Y. Zeng, Y. Zhong, B. Sheng, and Y. Zhang, "MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks," *J. Big Data*, vol. 10, no. 1, pp. 1–32, 2023, doi: 10.1186/s40537-023-00814-4.

[7] S. Mohanty *et al.*, "Detection of cyber threats from suspicious URLs using multi-classification approach," *Sustain. Sci. Intell. Technol. Soc. Dev.*, no. January, pp. 107–129, 2023, doi: 10.4018/979-8-3693-1186-8.ch007.

[8] I. Mumbai, Maharashtra, "Blended Malware Image Dataset," 2022, [Online]. Available: https://www.kaggle.com/datasets/gauravpendharkar/blended-malware-image-dataset

[9] M. Hu, J. Zhang, L. Matkovic, T. Liu, and X. Yang, "Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions," *J. Appl. Clin. Med. Phys.*, vol. 24, no. 2, pp. 1–21, 2023, doi: 10.1002/acm2.13898.

[10] R. Eriksson, "Deep Reinforcement Learning Applied to an Image-Based Sensor Control Task," 2021.

[11] P. Jaroensiripong, K. Sumongkayothin, P. Siritanawan, and K. Kotani, "Cybersecurity Intrusion Detection with Image Classification Model Using Hilbert Curve," *Proc. Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 2, pp. 325–332, 2024, doi: 10.5220/0012306100003660.

[12] Y. Peng, J. Zhang, and Z. Ye, "Deep Reinforcement Learning for Image Hashing," *IEEE Trans. Multimed.*, vol. 22, no. 8, pp. 2061–2073, 2020, doi: 10.1109/TMM.2019.2951462.

[13] J. Stember and H. Shalu, "Deep reinforcement learning-based image classification achieves perfect testing set accuracy for MRI brain tumors with a training set of only 30 images," Feb. 2021, [Online]. Available: http://arxiv.org/abs/2102.02895

[14] S. H. Oh, M. K. Jeong, H. C. Kim, and J. Park, "Applying Reinforcement Learning for Enhanced Cybersecurity against Adversarial Simulation," *Sensors*, vol. 23, no. 6, 2023, doi: 10.3390/s23063000.

[15] A. Schwaninger and A. Bolfing, "The impact of image-based factors and training on threat detection performance in X-ray screening," *Proc. 3rd ...*, no. October 2015, pp. 317–324, 2008, [Online]. Available: http://www.casra.ch/uploads/tx_tvpublications/SchBolHalHelBelHay2008_02.pdf