Investigating the Efficacy of Deep Reinforcement Learning Models in Detecting and Mitigating Cyber-attacks: a Novel Approach

S. Phani Praveen*[1], Anuradha Chokka[2], Pappula Sarala[3], Rajeswari Nakka[4], Suresh Babu Chandolu[5], V. Esther Jyothi[6]

[1]Department of CSE, PVP Siddhartha Institute of Technology, Vijayawada, A.P, India
[2]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India
[3]Department of CSE, Lakireddy Bali Reddy College of Engineering, Mylavaram, AP, India
[4]Department of Computer Science and Engineering, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
[5]Department of CSE, Dhanekula Institute of Engineering and Technology, Gangur, Vijayawada, A.P, India
[6]Department of Computer Applications, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, A.P, India
Emails: phani.0713@gmail.com; dranuradha@kluniversity.in; saralapappula05@gmail.com; rajeswari.gec@gmail.com; suresh.chandolu@gmail.com; vejyothi@vrsiddhartha.ac.in
* Corresponding Author: phani.0713@gmail.com

Abstract

Ordinary defence components like rule-based firewalls and mark based detection are not staying aware of the always expanding intricacy and frequency of cyber security dangers. The reason for this work is to explore the way that deep reinforcement learning (DRL), a subfield of artificial intelligence famous for its viability in handling testing decision-production situations, may be utilized to improve cyber security conventions. To mimic and balance threatening cyber-attacks, we present a system that utilizes deep reinforcement learning (DRL). We propose a specialist based model that can learn and adjust ceaselessly in powerful network security situations. In light of the present status of the network and the rewards it gets for its decisions, the specialist concludes what the best game-plans are. Specifically, we utilize the policy gradient (PG)- based double deep Q-network (DDQN) model and trial on three different datasets: NSL-KDD, CIC-IDS, and AWID. Our review demonstrates the way that DRL can really further develop the detection after-effects of cyber-attacks. Utilizing the policy gradient DDQN model on different datasets, we find prominent upgrades in cyber security conventions. Specific boundary modifications upgrade the viability of our philosophy much more, displaying empowering results on different datasets. This exploration features the potential of deep reinforcement learning (DRL) as a successful instrument in the field of cyber security. Our examination progresses detection techniques and gives a versatile arrangement that can be applied to an assortment of cyber security worries by giving areas of strength for a to demonstrating and relieving cyber dangers.

Keywords: Deep reinforcement learning; Detection; Cyber-attacks; Network security; Double deep Q-network; Policy gradient.

1. INTRODUCTION

It is common practice to employ red group practices while testing the resilience of network systems to different types of cyber attacks.

In order to test the system's defences against various tactics, plans, and techniques used by sophisticated attackers, these activities frequently use opponent profiles to mimic modern tireless attacks [1]. Red group activities can be exorbitant and tedious to do, however, on the grounds that they require particular human information. To make red joining more viable, emulators have been created to computerize and accelerate the attack re-enactment process [2].

The rising refinement and dynamic nature of cyber attacks has prompted an expanded significance of cyber attack re-enactment. Oftentimes, predefined rules and marks utilized in traditional security systems are insufficient to fight off these shrewd and versatile assailants. Then again, machine learning (ML) models have formed into a more adaptable and versatile strategy for cyber security [3]. With time, machine learning models can gain from verifiable information to turn out to be more skilled at distinguishing and tending to new risks. They could give a stronger defence instrument that adjusts to moving dangers [4]. The utilization of ML models in cyber security has exhibited significant commitment for working on the security of networked systems [5].

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

Creating independent cyber system security methodologies and activity suggestions is a difficult undertaking in the genuine world. Coordinating the elements among aggressors and safeguards and progressively defining the vulnerability in the system state are important to give decision help to cyber system security systems [6]. A branch of machine learning known as reinforcement learning (RL) can investigate and take advantage of novel and changing environments in order to learn from past experiences, making it a promising candidate for solving this problem [7]. RL gives defenders enough tools to carry out the best possible sequence of actions with the least amount of pre-knowledge about the attacker or the environment [8]. The defender can record different defensive and protective actions, using reinforcement learning approaches, in a variety of system states, and both discrete and continuous states in a high-dimensional space [9]. Given the complexity and quick spread of cyber attacks, reinforcement learning is a good fit for cyberspace. We evaluate the findings against a number of accepted standards, including recall, F1 score, accuracy, and precision. Our suggested framework, which makes use of the DDQN approach, performs better than comparable and contemporary models, according to the comparative results. This study's fundamental commitments are to cause to notice these benefits and offer them as a solid substitute for conventional ML models.

1.1. Deep Reinforcement Learning

A state of the art way to deal with artificial intelligence, deep reinforcement learning (DRL) mixes deep brain networks with reinforcement learning. Specialists can learn modern strategies by iteratively cooperating with the climate and settling on choices that amplify aggregate rewards [10]. DRL utilizes the force of deep learning to extricate complex highlights from unstructured information with the goal that specialists can gain runs straightforwardly from tactile data sources. Q-learning, policy gradient techniques, and entertainer pundit systems assume a significant part in DRL [11]. Esteem networks, policy networks, and compromises among investigation and abuse are significant ideas. DRL has a few applications in fields like mechanical technology, gaming, banking, and clinical [12].

Its development from Atari games to genuine difficulties features its flexibility and strength. Difficulties incorporate example adequacy, exploratory techniques, and security concerns. The association guarantees an inventive future that will change how decisions are made and issues are settled, with the objective of progressing DRL mindfully [13].

1.2. Overview of DRL in Cyber Security

Numerous applications of reinforcement learning to diverse facets of cyber security, spanning from safeguarding vital infrastructure to preserving data privacy, have been put forth in the literature [14]. Nevertheless, the limitations of conventional reinforcement learning have limited its potential to address intricate and extensive cyber security issues. With more and more gadgets becoming online, hacks have gotten more common and more sophisticated in recent years [15]. Through the integration of RL and deep learning, a new class of DRL methods has been developed to identify and counteract complex cyber-attacks such as malware, jamming, spoofing, intrusions into host computers or networks, distributed denial-of-service attacks, deception attacks against autonomous systems, intrusions into cyber-physical systems, and attacks in adversarial networking environments [16].

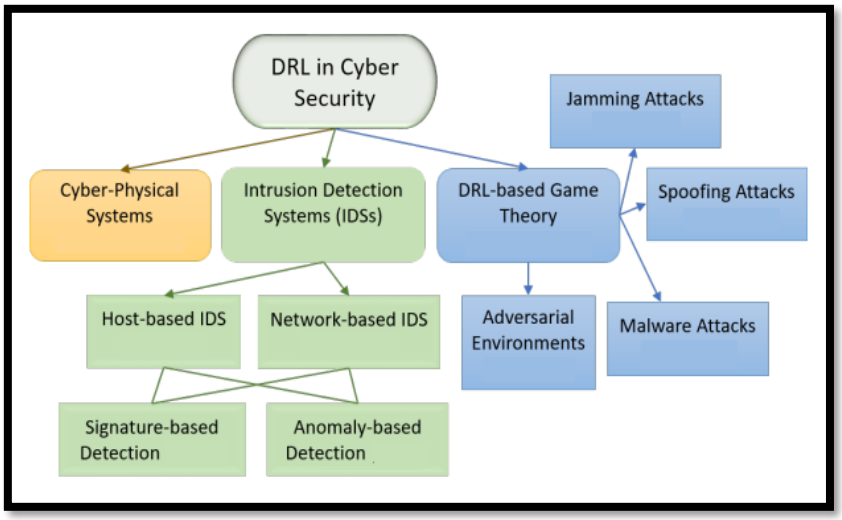


Figure 1: DRL in cyber security

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

- **DRL-based Cyber-Physical System Security Techniques**

Cyber security research community interest in and attention to protection mechanisms for cyber-physical systems (CPS) against cyber attacks has grown significantly. CPS is a system that is managed using internet-integrated computer-based algorithms [17].

The proliferation of control technology and the Internet has led to widespread usage of CPSs in many industries, including smart grid, manufacturing, health monitoring, transportation, and many more [18]. Cybercriminals target these systems more frequently due to their high Internet visibility.

- **DRL-based Intrusion Detection Systems**

In order to spot suspicious activity and identify intrusions, security experts usually have to keep an eye on and evaluate audit data. The quantity of audit data, however, grows exponentially with an increase in the size of the network [19]. That is why it is very difficult, if not impossible, to identify manually. An intrusion detection system (IDS) is a stage, either software or equipment, that is introduced on have PCs or network equipment [20]. Its goal is to identify and report any dubious or unsafe way of behaving to the executive by assessing review information [21].

- **DRL-based Game Theory for Cyber Security**

When it comes to cyber defense, the standard tools like intrusion detection systems, firewalls, and antivirus software tend to be static, one-sided, and slow to respond to changing threats. In order to offer reliable cyber security, it is important to consider the interplay between the various cyber components present in cyberspace [22]. To be more exact, the security strategy applied to one part influences the choices made by different parts. Consequently, the choice space expands considerably and incorporates a few consider the possibility that situations when the framework is enormous [23]. Since game hypothesis can examine different circumstances and decide the ideal system for every member, it has been demonstrated to be viable in tackling such huge scope issues. A game player's utility or reward is influenced by both its actions and those of other players [24]. Stated differently, the effectiveness of cyber defense tactics needs to consider the tactics used by attackers as well as the actions of other users on the network.

1.3. Objectives of the study

The primary aims of this study are outlined below:

- To assess the effectiveness of DRL models like DDQN with PG in detecting and mitigating cyber attacks across various datasets.
- To explore the adaptability of DRL in dynamic security environments.
- To optimize DRL parameters to enhance cyber-attack detection outcomes.

2. REVIEW OF LITERATURE

Researchers look at the utilization of deep reinforcement learning (DRL) to shield cyber systems from eccentric and dynamic antagonistic movement [25]. In this study the creators stress how versatile DRL is and the way that it tends to be utilized to learn and change in accordance with new weaknesses ceaselessly. They contrast this with regular defences, which are as often as possible rule-based and inflexible and are consequently improper for dynamic danger situations.

They propose a structure that utilizes DRL to get familiar with the best security strategies by means of connection with the environmental elements. With this technique, the system can distinguish and foil refined dangers continuously. The work puts areas of strength for an on preparing a DRL specialist in a virtual setting that repeats genuine cyber threats. The expert can change its defences because of attacks as a major characteristic, showing critical improvement in its preventiveness above static techniques [26].

Researchers permit intrusion detection systems (IDS) to work on their viability and accuracy by fixing the issue of choosing the most important features [27]. In unique network environments, standard part choice cycles often come up short. They give a unique DRL-put together viewpoint with respect to independent part choice as an answer for this issue. A Markov decision process (MDP) model is utilized to depict the system, and the DRL specialist picks elements to upgrade IDS execution. Deep brains networks are utilized by the specialist to deal with high-layered include spaces after it was prepared with a prize component that rewards high detection exactness and low misleading up-sides. Powerful tests directed on reference datasets show outstanding improvements in detection accuracy and adequacy [28].

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

Authors inspected at the 38th Yearly Gathering on PC Security Applications. This study utilizes DRL to shrewd lattice networks, which are defenceless to cyber-attacks due to their interconnectedness [29]. Constant assault detection and independent network activity are made conceivable by the Winged serpent structure. The DRL specialist manages lattice exercises in this structure and watches out for any abnormalities that would highlight an attack. Generous attack recognition and effective grid working are changed by the honor instrument. The DRL expert arrangements with the dimensionality and complexity of the splendid grid environment utilizing deep mind networks. Re-enactments display how well Legendary snake controls network errands and can unequivocally recognize a variety of cyber-attacks [30].

Authors look at the safe control issue for cyber-physical systems (CPS) under counterfeit information infusion dangers utilizing profound support learning [31]. The safe regulator plan for CPS under attacks is figured out as an activity strategy picking up using information, in light of our depiction of the CPS under assaults as a Markov decision process (MDP). To prepare a safe strategy for CPS disconnected notwithstanding assaults, a Lyapunov-based delicate entertainer pundit learning approach is introduced. As opposed to past discoveries, this study exhibits the learning calculation's intermingling as well as the framework's dependability while utilizing the learnt strategy. This is critical for applications that rely upon security and soundness. The viability of the proposed conspire is at long last shown utilizing a robot arm framework and a satellite disposition control framework. Also, the upsides of the control calculation created in this paper are outlined through examinations between the proposed learning calculation and the traditional PD regulator [32].

Researcher's centres around safeguard methodologies to take the framework back to ordinary working circumstances during cyber-attacks [33]. For feeders that are impacted, an original deep reinforcement learning (DRL) strategy is made to limit voltage infringement and lower power misfortunes. The guard challenge is rehashed as a Markov dynamic interaction determined to limit load shedding and controlling DERs progressively. This is achieved by an improved soft actor-critic (SAC)- based DRL calculation, which utilizes the auto-tune entropy and Gaussian strategy attributes to control DER set focuses and load-shedding situations in discrete and persistent modes. Results from mathematical examinations between the proposed technique and other control draws near, including Volt-VAR (VV), Volt-Watt (VW), and model predictive control (MPC), on a changed IEEE 123-hub framework show the way that the proposed strategy can totally moderate cyber threats by taking out voltage infringement and giving serviceable control activities [34].

3. MATERIALS AND METHOD

3.1. Dataset Descriptions

3.1.1. NSL_KDD datasets

The effectiveness of the training model could be negatively affected by duplicate data in the conventional KDD99 network traffic dataset. Tavallaee et al. set out to solve this problem in 2009 by developing the NSL-KDD dataset. By changing how much information in the testing and training sets contrasted with the genuine dataset and erasing unessential data from the first dataset, the dataset upgrades the rationale of the KDD dataset and further develops model training execution. There are 24566 records in the test dataset and 43 features that can be categorized as normal or attack, compared to 146175 records in the training dataset. Four types of assaults are included in the dataset: DOS, probing, R2L, and U2R. Using the NSL-KDD dataset, we use DDQN models in this study to obtain key performance measures for attack detection.

3.1.2. AWID datasets

The Aegean Wi-Fi Intrusion dataset (AWID) is available to the public and comprises typical network traffic as well as three assaults on IEEE 802.11 networks. Out of all the datasets offered by AWID, we have selected the AWID-CLS-R dataset because it provides different datasets for training and testing. Four classification categories are included in this dataset: impersonation, flooding, injection, and normal. It has 156 features, 1997596 training samples, and 595844 testing samples. We diminished the quantity of attributes to 26 by eliminating those with consistent or invalid values as well as those with test explicit network tends to that didn't relate to the test information. We have encoded the clear cut qualities and standardized the consistent information to a scope of 0 to 1. Regardless of having a similar name circulation for the training and testing datasets, the AWID dataset displays a significantly more serious level of irregularity than the NSL-KDD dataset. Thusly, grouping calculations have various deterrents while managing the AWID and NSL-KDD datasets. Here, we apply the DDQN model to the AWID dataset and assess it utilizing key execution pointers, like recall, accuracy, precision, and F1 score, with an emphasis on assault recognition.

3.1.3. CSE-CIC-IDS datasets

University of New Brunswick scientists gathered a dataset for DDoS information assessment that included seven unmistakable attack situations. The dataset consists of 18,000,000 occurrences that were gathered over a period of ten days. Content-wise, it consists of ninety characteristics that are extracted from network traffic and computer system logs

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

using CICFlowMeter-V3. This study's DDQN models use the CSE-CIC-IDS dataset to evaluate basic execution measurements in recognizing ordinary and odd mark values.

3.2. Model Description

Approximators are utilized by profound learning models, as brain organizations, to address the approaches and worth capabilities in support learning. The hypothesis of Markov decision process (MDP) structures the premise of support learning. The likelihood dissemination of transforming from the current state (s) to new state (s0) is addressed by the transition function (T). Each state-activity blend has an outright worth given by the reward function (R), and the worth of the discount factor (g), which goes from 0 to 1, shows the significance of potential compensations. In a MDP, the transition function (T) fulfils the Markov property, implying that the activity and present status alone decide the likelihood of changing to another state, autonomous of past events. After a Markov decision process (MDP) has been found, a policy is made to link each state to an action, with the aim of finding the optimal policy that maximizes the expected total rewards. One can quantify the optimality criterion by adding up all of the benefits, averaging them, or using a discount rate to give priority to instant rewards. An MDP offers an agent's theoretical foundation for decision-making and interaction with its surroundings. While the agent carries out the policy, the environment handles the transition and reward functions. Usually, the agent's interaction with the environment is broken down into distinct time steps. By constructing a value function that approximates the value connected to every state, the relationship between the optimality criterion and policy is formed. Under this case, we directly learn the most efficient policy. Policy gradient-based techniques allow us to calculate the transition probabilities between states.

3.2.1. Proposed models

DDQN performs better than DQN in the RL domain. It tackles a crucial problem with DQN: the tendency to overestimate action values, which can result in bad choices. The problem with DQN is that it uses a single neural network to estimate the Q-values for every action and condition. Based on past experiences, this network changes the Q-values and decides what to do next. Because of this relationship, there may be an overestimation bias, particularly for infrequently chosen actions, where the network overestimates Q-values. This could lead to the agent acting badly more often. DDQN makes use of two different networks: The Q-values for each activity in a state are determined by the main network. The target network acts as a "frozen" version of the main network, gradually updating its parameters. Estimation and action selection are two different processes. The action with the greatest Q-value in each state is selected by the main network to be the subsequent action. Nonetheless, the decision is made to update the Q-values of the target network with those of the main network. This eliminates the direct impact of the updating procedure on perhaps inflated Q-values.

3.2.2. Models details

Through action, reward monitoring, and state prediction, the agent engages with its environment. A replay buffer is used to store these encounters. A Q-function estimates the maximum possible reward given a set of inputs, such as an activity and an environment. Therefore, the states and actions combinations determine Q(s, a). After the Q-function is constructed, the policy function, which dictates the actions to be taken in each state, may be deduced. Here is the process by which the state-dependent policy function is generated from the Q-function:

$$\boldsymbol{Policy(s)} \;=\; \boldsymbol{arga\,max}\,(\boldsymbol{Q(s,a)}) \qquad \textbf{(1)}$$

A training method known as an epsilon-greedy policy allows the agent to investigate possible courses of action and identify the optimal course of action as the number of explorations increases. If one wants to predict an action, they can use the probability (1 − e) or they can choose it randomly using the probability ε. By applying an MDP solution that maximizes the potential lower reward in every state, the optimal policy (π*) is found. The special features and presence of the fixed-point answer to the ideal Bellman issue are described by the following formula.

$$\mathbf{Q} * (\mathbf{a}, \mathbf{s}) \;+\; \mathbf{R}\,(\mathbf{a}, \mathbf{s}) \;+\; \mathbf{g} \int_{\mathbf{s}'} \mathbf{T}\,(\mathbf{S}'/\mathbf{s},\, \mathbf{a})\, \mathbf{maxQ} * \,(\mathbf{a}', \mathbf{s}') \qquad \textbf{(2)}$$

Through proper optimization, the Q-value's performance is improved. The procedure starts with a standard sample that contains the following states: the next state (st+1), the present state (st), and the appropriate label for that state (at). A mini-batch is a selection of samples taken at random from the larger dataset, of which this sample is simply one of several. Random samples from the dataset are used to create a new mini-batch for each training cycle. Each training cycle will conclude with the creation of a new mini-batch, which will process the samples. Figure 2 shows the iterative process as a bold arrow representing the vector of values Q(st, {a}).

$$\mathbf{Q}(\boldsymbol{s_t}, \{\mathbf{a}\}) \;=\; [\mathbf{Q}(\boldsymbol{s_t}, \{\mathbf{a}\}\mathbf{0}), \mathbf{Q}(\boldsymbol{s_t}, \{\mathbf{a}\}\mathbf{1}), \ldots\, \mathbf{Q}(\boldsymbol{s_t}, \{\mathbf{a}\}\mathbf{p})] \qquad \textbf{(3)}$$

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

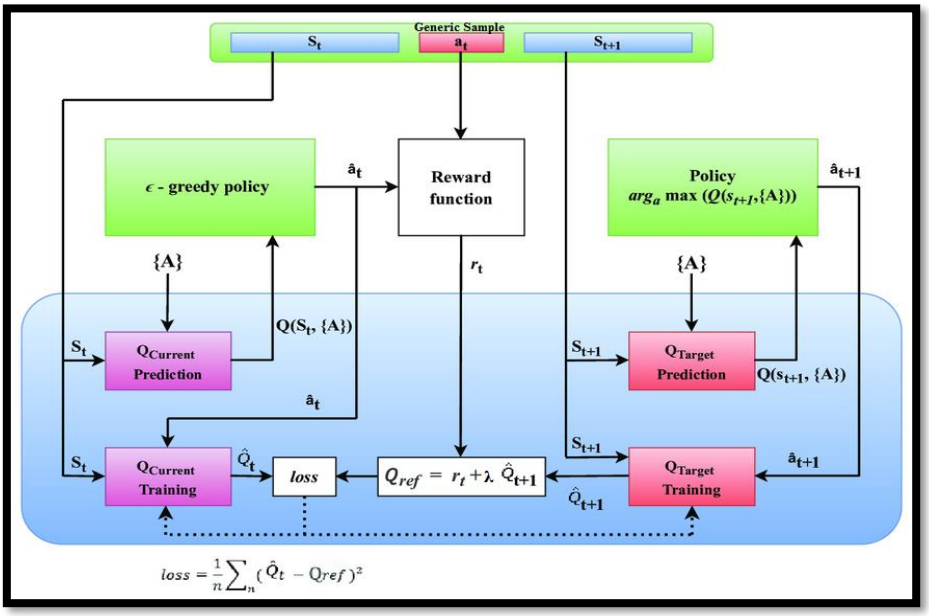


Figure 2: The DDQN model

The environment in which the datasets are used is managed by the DDQN agent. Setting the model parameters and the algorithm's initial values is the first step. The DDQN state variables (s) are represented by the values of the dataset features (F1–Fm). It is important to remember that the DDQN method has a batch size (bs) of 100. This indicates that each state reads 100 dataset records into memory and assigns them to a single state. However, there are a lot of state variables, and they can all have different values. Maintaining state-value pairings in a Q-table is harder as the number of pairs rises.

To inexact Q values from actions and states, the DDQN specialist utilizes a neural network. The preparation test, or Batch n, gives the state (sn) at each discrete state, as found in Figure 3.

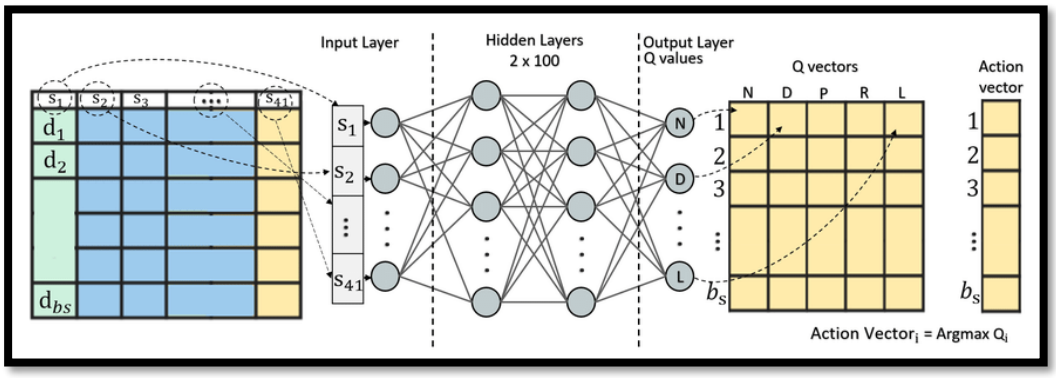


Figure 3: A deep neural network and states are used to generate predictions by the DDQN model

3.3. Double deep Q-network (DDQN) algorithm

The Double Deep Q Network is the Double Q-Learning implementation using a Deep Neural Network (Double DQN). H. van Hasselt proposes double DQN in 2016. Double DQN, which takes inspiration from Double Q-Learning, employs the Deep Q Network (DQN) and Target Network, two distinct Deep Neural Networks. A sophisticated reinforcement learning technique called the Double Deep Q-Network (DDQN) algorithm was created to overcome the overestimation bias present in conventional Q-learning techniques, especially when it comes to cyberattack detection and mitigation. Developing crafted by Deep Q-Networks (DQN), DDQN utilizes two unmistakable brain organizations to lessen the chance of activity esteem misjudgment: an objective organization for assessing Q-esteem and an essential organization for choosing activities (Vinayakumar, 2019). The program works by connecting with the climate iteratively, picking activities to adjust investigation and abuse utilizing an epsilon-voracious strategy, and refreshing the brain network loads in light of replay memory-put away encounters. Assessment estimates like accuracy, F1-score, precision, and recall show

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

that DDQN further develops dependability and execution in definitively distinguishing and relieving interruptions through the most common way of learning suitable guidelines for digital protection occupations.

Steps for DDQN execution:

Step 1: Experience Replay Development - Reinforcement learning utilizes the Experience Replay strategy to assist specialists with gaining from an assortment of encounters and recollections.
Step 2: Building the Target Network - The objective of the objective organization is to figure the assessed Q-esteem utilizing an activity that was browsed the Deep Q network (DQN).
Step 3: Form a Deep Q Network (DQN) - DQNs are utilized to figure out which game-plan has the most elevated Q-value in a specific condition.
Step 4: Specialist Development - Double Q Organization will be utilized to show the specialist all alone.
Step 5: Start Training.

Algorithm 1: Double Deep Q-Network (DDQN)

```
# Import necessary libraries
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from collections import deque
import random

# Define parameters
STATE_SIZE = 43  # Number of features in the NSL-KDD dataset
ACTION_SIZE = 5  # Number of possible actions (attack categories)
BATCH_SIZE = 100
EPISODES = 1000

# Define DDQN model
def build_model():
    model = Sequential()
    model.add(Dense(64, input_dim=STATE_SIZE, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(ACTION_SIZE, activation='linear'))
    model.compile(loss='mse', optimizer=tf.keras.optimizers.Adam(lr=0.001))
    return model

# Define DDQN agent
class DDQNAgent:
    def __init__(self):
        self.model = build_model()
        self.target_model = build_model()
        self.replay_memory = deque(maxlen=2000)
        self.gamma = 0.95  # Discount factor
        self.epsilon = 1.0  # Exploration rate
        self.epsilon_min = 0.01
        self.epsilon_decay = 0.995

    def remember(self, state, action, reward, next_state, done):
        self.replay_memory.append((state, action, reward, next_state, done))

    def choose_action(self, state):
        if np.random.rand() <= self.epsilon:
            return np.random.randint(ACTION_SIZE)
        q_values = self.model.predict(state)
        return np.argmax(q_values[0])

    def replay(self):
        if len(self.replay_memory) < BATCH_SIZE:
            return
```


Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

```
        minibatch = random.sample(self.replay_memory, BATCH_SIZE)
        for state, action, reward, next_state, done in minibatch:
            target = self.target_model.predict(state)
            if done:
                target[0][action] = reward
            else:
                t = self.model.predict(next_state)[0]
                target[0][action] = reward + self.gamma * np.amax(t)
            self.model.fit(state, target, epochs=1, verbose=0)

    def target_train(self):
        weights = self.model.get_weights()
        target_weights = self.target_model.get_weights()
        for i in range(len(target_weights)):
            target_weights[i] = weights[i]
        self.target_model.set_weights(target_weights)

    def train(self, env):
        for episode in range(EPISODES):
            state = env.reset()
            state = np.reshape(state, [1, STATE_SIZE])
            done = False
            total_reward = 0
            while not done:
                action = self.choose_action(state)
                next_state, reward, done, _ = env.step(action)
                next_state = np.reshape(next_state, [1, STATE_SIZE])
                total_reward += reward
                self.remember(state, action, reward, next_state, done)
                state = next_state
                self.replay()
                self.target_train()
            if self.epsilon > self.epsilon_min:
                self.epsilon *= self.epsilon_decay
            print("Episode: {}, Total Reward: {}".format(episode+1, total_reward))

# Define the environment for training
class Environment:
    def __init__(self):
        # Initialize NSL-KDD dataset and other necessary variables
        pass

    def reset(self):
        # Reset environment to initial state
        pass

    def step(self, action):
        # Perform action in the environment and return next state, reward, done flag, and additional info
        pass

# Create DDQN agent and environment
agent = DDQN Agent()
env = Environment()

# Train the agent
agent.train(env)
```

4. EXPERIMENTAL RESULTS

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

We tested the suggested DRL-based DDNQ model on three datasets: AWID, CIC-IDS, and NSL-KDD. It is imperative to emphasize that we employed the model containing two and three layers and performed a comparative analysis of them. To assess the various models' prediction performance, we provide the following performance metrics: recall, accuracy, precision, and F1 score. Our definitions of these performance metrics are based on accepted norms. Because the datasets are uneven, we will place more weight on accuracy and F1 score.

4.1. Performance metrics

We use a number of criteria, to assess the effectiveness of the DDQN model in identifying network assaults. The accuracy values alone, however, are not sufficient to judge the model's success because it also evaluates the proportions of correctly classified samples. The incorrectly classified samples are ignored. Data on true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) were used to establish these metrics. The attack and valid vectors that were mistakenly identified were categorized as FN and FP, respectively. The number of real attack vectors that were correctly categorized is shown by the TN and TP.

Accuracy: The accuracy statistic measures how many of the model's predictions were accurate relative to the total number of forecasts. Equation (4) can be used to calculate the accuracy metric.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TN+TP}}{\boldsymbol{TN+FN+TP+FP}} \times \mathbf{100} \quad \textbf{(4)}$$

Precision: It is a measurement of the proportion of positive cases compared to the total number of anticipated positive cases. Equation (5), which defines accuracy, is used to determine the correctness of the model.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+FP}} \times \mathbf{100} \quad \textbf{(5)}$$

Recall: Ratio of the total number of favorable occurrences. Equation (6), which shows this metric, shows how many accurate cases the model ignored when providing correct instances.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \times \mathbf{100} \quad \textbf{(6)}$$

F1 score: The accuracy and recall scores are averaged to determine the performance metric. It takes into account what each value has to offer. Recall and precision values are the basis for the F1 score, as seen by equation (7).

$$\boldsymbol{F1-score} = \mathbf{2} \times \frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Recall+ Precision}} \times \mathbf{100} \quad \textbf{(7)}$$

4.2. Results for NSL-KDD datasets

Showing the utility of DRL models for digital danger recognition in systems administration is a huge commitment of this work. Figure 4 shows the results of the DRL model review utilizing the NSL-KDD dataset.

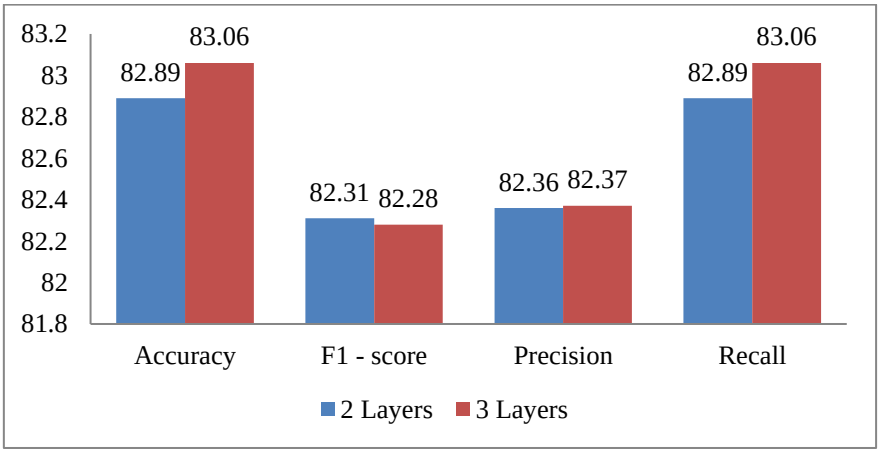


Figure 4: Results from the DDQN model for the NSL-KDD dataset

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

Applying the DDQN model to the NSL-KDD dataset at different network layer depths and comparing performance metrics between a 2-layer and a 3-layer configuration is shown in Figure 4. The outcomes show slight variations between the two configurations. At 82.89% and 83.06%, respectively, the 3-layer DDQN model slightly beats the 2-layer model in terms of accuracy and recall. Nevertheless, the F1-score of the 2-layer model is somewhat higher (82.31%) than that of the 3-layer model (82.28%), indicating a more balanced performance in terms of recall and precision. With the 2-layer model at 82.36% and the 3-layer model at 82.37%, the precision for both models is essentially the same. These findings imply that the model's performance is not considerably improved by going from two to three layer depths, suggesting that a more basic 2-layer model may be adequate for producing reliable detection results with the NSL-KDD dataset.

The NSL-KDD dataset presents different moves for classifiers because of the creation of the test and preparing sets. The measurements act as the essential proportions of execution for an attack detection framework, with the goal of accurately identifying the maximum number of attacks. Table 1 lists the DDQN model's evaluation metrics, broken down by class, using the NSL-KDD dataset.

Table 1: Class-by-class evaluation measures for the DDQN model using the NSL-KDD dataset

Metric	Attack Categories				
	Normal	**DoS**	**R2L**	**Probe**	**U2R**
Accuracy	88.63	95.79	88.96	96.65	99.83
F1-score	87.87	92.79	40.14	80.30	22.19
Precision	81.04	96.90	52.07	81.95	26.73
Recall	95.98	89.01	32.73	78.71	21.35

Table 1 provides an extensive analysis of the assessment metrics, broken down by attack types, for the performance of the DDQN model using the NSL-KDD dataset. The discoveries show varying levels of adequacy in recognizing and sorting different sorts of cyber attacks. Eminently, the precision of the DDQN model reaches from 88.63% for typical attacks to an astonishing 99.83% for U2R strikes, major areas of strength for showing execution. Comparative patterns should be visible in F1-scores, where higher qualities propose a superior harmony among review and accuracy. Review values demonstrate the way that well the model can perceive certified up-sides, while accuracy values demonstrate the way that well it can decrease misleading up-sides, with DoS and Test attacks showing particularly high accuracy scores. These outcomes suggest that the DDQN model effectively separates between different assault types, achieving significant exactness and accuracy all around, which is fundamental for solid network protection safeguard strategies.

4.3. Results for AWID database

Utilizing the AWID data set, the ongoing review looked to assess the adequacy of the RL model in cyber attack detection. A few order measurements, including as review, exactness, accuracy, and F1-score, were utilized. Figure 5 shows how well the models performed.

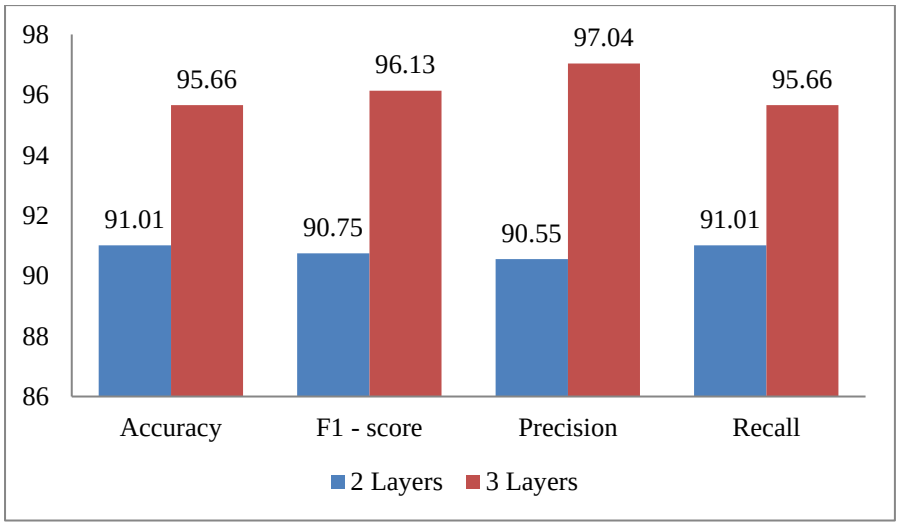


Figure 5: The results of the DDQN model for the AWID dataset

The exhibition after-effects of utilizing the DDQN model on the AWID dataset at two particular network layer profundities — two layers and three layers — are displayed in Figure 5. Updating the layer profundity from a few fundamentally further develops all presentation markers, as per the information.

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

The precision of the 3-layer DDQN model is 95.66%, which is fundamentally more prominent than the 2-layer model's 91.01%. Along these lines, the 3-layer model's F1-score of 96.13% is essentially higher than the 2-layer model's 90.75%. Critical enhancements are additionally found in review and accuracy, with the 3-layer model getting review of 95.66% and accuracy of 97.04% rather than 90.55% and 91.01% for the 2-layer model. These discoveries suggest that the additional layer extraordinarily works on the model's ability to perceive and sort cyber threats in the AWID dataset, proposing that a more many-sided network construction can all the more likely address unpredictable examples and connections in the information. Table 2 shows the class-by-class assessment measurements of the DDQN model utilizing the AWID dataset.

Table 2: Class-by-class assessment measurements of the DDQN model using the AWID dataset

Metric	Attack Categories			
	Normal	Impersonation	Injection	Flooding
Accuracy	95.69	96.86	99.31	99.48
F1-score	97.96	70.71	93.38	83.69
Precision	99.28	56.86	86.83	84.85
Recall	95.74	93.68	99.99	82.56

Table 2 offers a careful examination of assessment measurements, separated by assault type, for the DDQN model's presentation utilizing the AWID dataset. For every one of the attack classifications — Ordinary, Pantomime, Infusion, and Flooding — the estimations incorporate accuracy, F1-score, precision, and recall. The results show how well the model recognized and ordered the different sorts of cyber attacks found in the AWID dataset. Across all assault classes, the DDQN model remarkably shows incredible accuracy, going from 95.69% for Typical attacks to 99.48% for Flooding attacks. F1-scores show a harmony among recall and precision, with Ordinary attacks scoring especially high (97.96%) and Infusion attacks scoring 93.38%. The accuracy values exhibit how well the model lessens misleading up-sides; prominent precision scores for Ordinary attacks (99.28%) and Infusion attacks (86.83%) are noticed. Recall values, then again, demonstrate the way that well the model can identify genuine up-sides, with high scores for pantomime attacks (93.68%) and infusion attacks (99.99%). These outcomes represent the viability of the DDQN model in exactly distinguishing and classifying various sorts of attacks inside the AWID dataset, highlighting its true capacity for solid digital protection guard strategies.

4.4. CIC_IDS database results

The survey analyzed the after-effects of applying the DRL model to the CIC_IDS dataset on the fourth day of the dataset. The examination's decisions showed that the model yielded great results, as displayed in Figure 6.

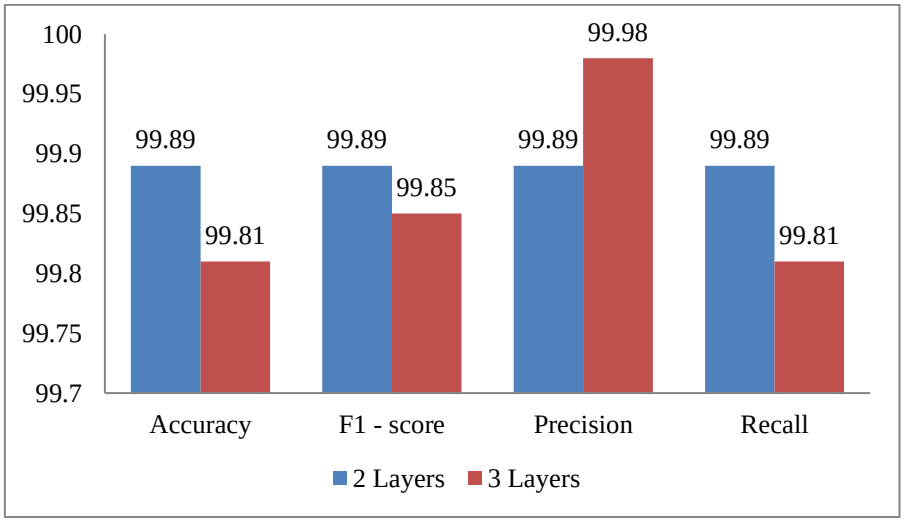


Figure 6: Results of DDQN model for CIC_IDS dataset

The DDQN model's exhibition on the CIC_IDS dataset is shown in Figure 6, which differentiates a 2-layer network with a 3-layer network. With simply little varieties, the two plans perform strikingly well by all actions. The 2-layer model performs immaculately, with a 99.89% accuracy, F1-score, precision, and recall. With a F1-score of 99.85% and a

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

minuscule decrease in accuracy and recall at 99.81%, the 3-layer model keeps on performing surprisingly well. However, as far as accuracy, it performs insignificantly better compared to the 2-layer model, hitting an almost immaculate 99.98%. As indicated by these discoveries, the CIC_IDS dataset can be successfully modelled utilizing both the 2-layer and 3-layer DDQN models. The 2-layer model offers insignificantly better generally speaking consistency, while the 3-layer model offers essentially higher precision. This proposes that the less complex 2-layer model performs similarly too for this particular dataset and that rising the layer profundity doesn't enormously work on generally speaking model execution. Table 3 shows the class-by-class assessment measurements of the DDQN model utilizing the CIC_IDS dataset.

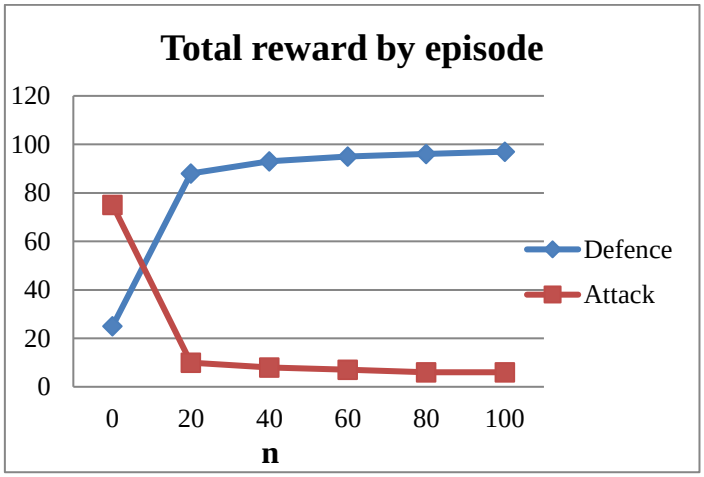
Table 3: Class-by-class breakdown of the DDQN model's assessment measurements utilizing the CIC_IDS dataset

Metric	Attack Categories	
	Benign	DDoS
Accuracy	99.80	99.80
F1-score	99.36	93.36
Precision	99.99	86.81
Recall	99.71	99.99

Table 3 gives an exhaustive examination of the evaluation measurements for the presentation of the DDQN model utilizing the CIC_IDS dataset, separated into two classifications: disseminated disavowal of administration (DDoS) and harmless attacks. The results show that the DDQN model performs uncommonly well in both assault classes. Specifically, the model effectively groups cases inside the dataset, showing its surprising accuracy appraisals of 99.80% for both DDoS and harmless attacks.

F1-scores, which show a decent harmony among recall and precision, likewise show great execution, with DDoS attacks scoring 93.36% and harmless attacks scoring 99.36%. The model's precision values, which demonstrate its capacity to lessen misleading up-sides, are extremely imperative. It accomplishes practically impeccable scores of 99.99% for harmless attacks and 86.81% for DDoS attacks. Recall scores, which demonstrate the way that well the model can recognize genuine up-sides, are likewise astoundingly high, coming in at 99.71% for harmless attacks and 99.99% for DDoS attacks. These discoveries exhibit the DDQN model's uncommon viability in exactly distinguishing and separating among unsafe and harmless traffic in the CIC_IDS dataset, featuring its true capacity for dependable network protection applications.

The growth of reward and loss values during the preparation of a DDQN model in the cyber security environment is portrayed in Figures 7, 8, and 9.



Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

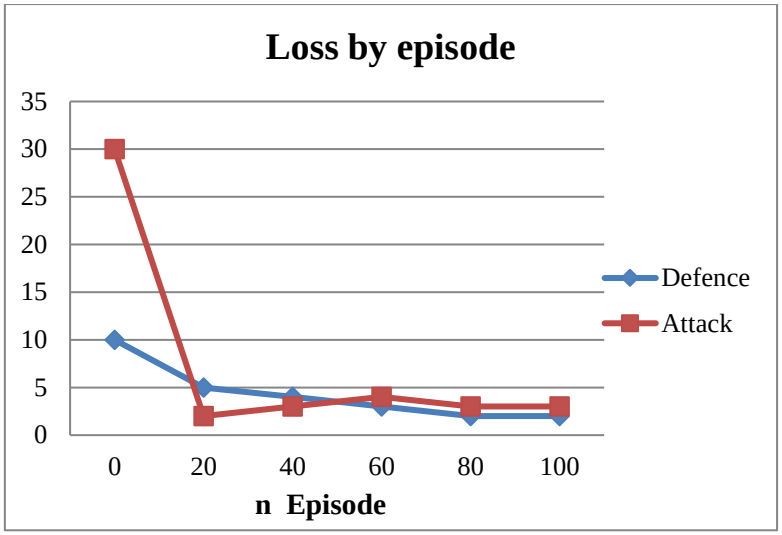
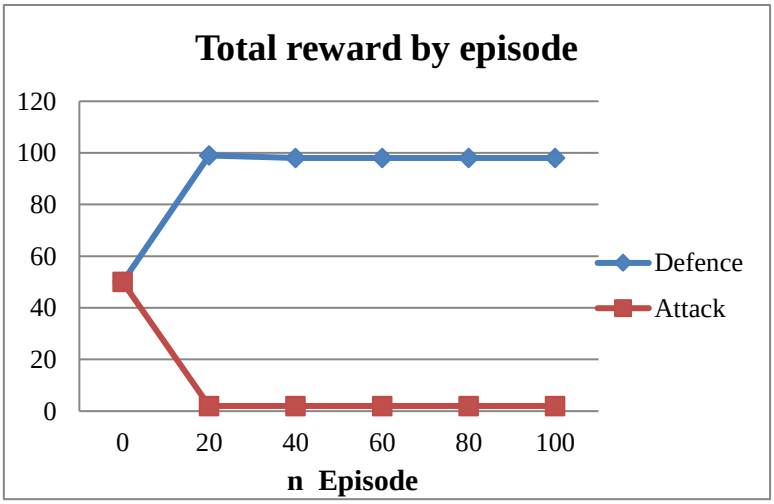


Figure 7: Across NSL-KDD Datasets, environment preparing to reward and loss values achieved during DDQN model preparation



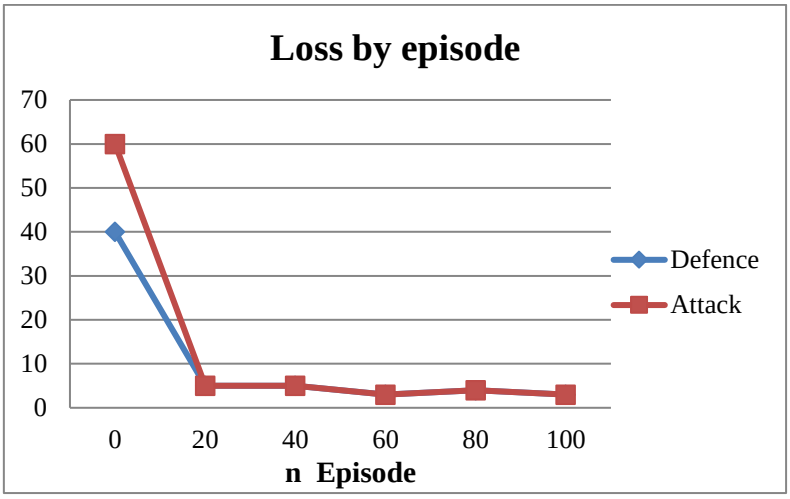


Figure 8: Preparing the environment to reward and lose factors got from preparing the DDQN model on different CIC-IDS datasets

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

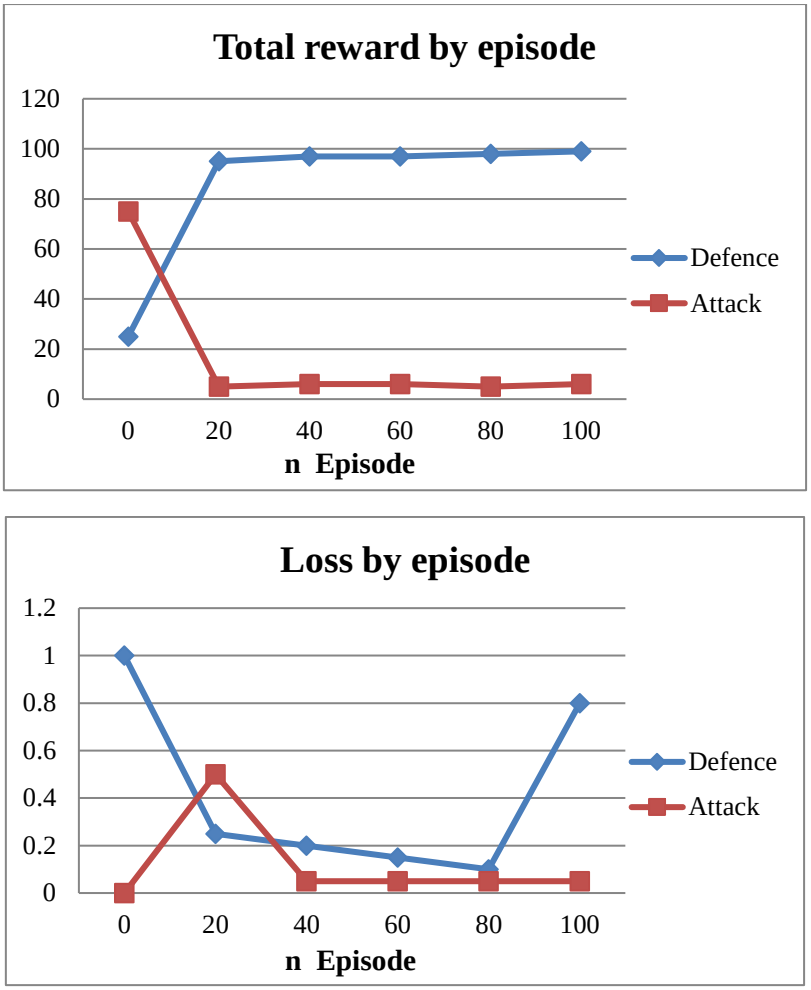


Figure 9: Utilizing AWID Datasets, the environment is prepared to reward and lose values accomplished during DDQN model preparation

The improvement of all out rewards and loss values is displayed in Figures 7, 8, and 9 when a DDQN model is prepared on the NSL-KDD, CIC-IDS, and AWID datasets, separately. The outcomes exhibit that safeguard specialists' all out rewards in all datasets expansion in a steady way with an expansion in training episodes, proposing that their presentation in moderating cyber attacks has expanded. As an outline of the model's viability against attacks, consider the NSL-KDD dataset, where the safeguard specialist's all out reward starts at 25 and increments to 97 by the 100th episode, while the aggressor's payout falls decisively from 75 to 6. Similarly, in the CIC-IDS dataset, the reward for the safeguard specialist ascends from 50 to 98, while the reward for the assailant tumbles from 50 to 2. Comparative patterns should be visible in the AWID dataset, where the instalment for the protection specialist increments from 25 to 99, while the payout for the assailant diminishes from 75 to 6. In tandem, the safeguard specialist's loss values decline, connoting worked on model soundness and learning viability. For example, in the NSL-KDD dataset, the loss diminishes from 1.00 toward the starting to 0.08 by the 100th episode. With additional training, the DDQN model turns out to be more proficient at battling off cyber attacks, as proven by greater rewards for protection specialists and lower loss values. This general pattern across all datasets shows a striking improvement in both execution and learning strength.

States, actions, and rewards in the training environment make up the reinforcement learning (RL).Stated differently; rewards are earned when an action is taken to go from state S1 to state S2. The action is right and the payoff is big when the defender counterattacks the attacker.

Table 4 shows how the RL models contrast with past examinations that were utilized to distinguish and characterize various kinds of attacks in the three data sets.

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

Table 4: Examination of the discoveries from the three datasets with the discoveries from past exploration

Method	Datasets	Accuracy (Acc)	F1-Score (F1)	Precision (Pre)	Recall (Rec)	Features
DDQN	NSL-KDD	90.80	92.04	90.46	94.05	124
	AWID	96.09	94.96	93.37	96.09	26
DQL	NSL-KDD	79.09	82.43	78.86	77.78	43
AE-RL	NSL-KDD	81.18	80.06	80.76	81.18	43
	AWID	96.11	97.28	98.04	96.11	26
DDQ N	AWID	97.49	97.75	98.06	50.02	-
	NSL-KDD	74.45	70.04	67.63	123.02	-
DQN	CIC_IDS	95.13	93.53	-	-	15
Our proposal (DDQN)	NSL-KDD	83.06	82.28	82.37	83.06	164
	AWID	95.66	96.13	97.04	95.66	69
	CIC_IDS	99.81	99.85	99.98	99.81	39

The presentation results of numerous reinforcement learning techniques are looked at in Table 4 utilizing three cyber security datasets. Across all datasets, the DDQN approach ceaselessly shows great execution. DDQN obtains 90.80% accuracy, 92.04% F1-score, 90.46% precision, and 94.05% recall for NSL-KDD. DDQN acquires 96.09% accuracy, 94.96% F1-score, 93.37% precision, and 96.09% recall in the AWID dataset. On contrast, DDQN performs outstandingly on the CIC_IDS dataset, showing 99.81% accuracy, 99.85% F1-score, 99.98% precision, and 99.81% recall. Then again, DDQN regularly beats different methodologies, for example, DQL and AE-RL, which show lower execution measurements in NSL-KDD. Moreover, when contrasted with prior techniques, the recommended DDQN model performs better on all datasets, showing its adequacy in cyber security assignments. These discoveries feature the DDQN approach's versatility and flexibility in handling cyber security issues across a scope of datasets.

5. Discussion

The review utilizes state of the art AI techniques to resolve the earnest issue of cyber security dangers. To improve cyber attack location and moderation, this study utilizes Deep Reinforcement Learning (DRL) models, to be specific the DDQN, on an assortment of datasets, like NSL-KDD, AWID, and CSE-CIC-IDS, every one of which has its own arrangement of highlights and issues.

The training and evaluation of the DDQN models depend on the datasets utilized in this review. Further developed model training execution and judiciousness are accomplished by eliminating repetitive information and changing training and testing information volumes in the NSL-KDD dataset, which is a superior variant of the KDD99 dataset. It consists of 24,566 test records and 146,175 training records with 43 attributes (U2R, R2L, DOS, and probe) that categorize records as normal or assaults. The AWID dataset, created to counteract IEEE 802.11 network attacks, consists of 1,997,596 training samples and 595,844 test samples, reduced to 26 characteristics. It also includes regular traffic and three different types of attacks (injection, flooding, and impersonation). The CSE-CIC-IDS dataset, which contains 18 million instances with 90 parameters, includes statistics on seven different attack methods over a ten-day period.

By reducing overestimation bias by employing two distinct networks—a target network for evaluating Q-values and a primary network for action selection—DDQN models improve on conventional Q-learning. This division aids in the provision of more precise Q-value updates, which improves the agent's ability to make decisions. Within the context of a Markov decision process (MDP), the DDQN model maximizes the expected total rewards while taking future benefits into account with a discount factor. This allows the model to learn how to map states to actions. Replay buffers ensure diversified training samples and prevent recent events from unduly influencing the learning process by storing the agent's actions and rewards.

The experimental findings demonstrate how well DDQN models work across datasets to identify and mitigate intrusions. While the 2-layer model has a slightly higher F1-score (82.31%), the 3-layer DDQN model performs somewhat better for the NSL-KDD dataset in terms of accuracy (83.06%) and recall (82.89%). The assessment metrics, which are segmented

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

by attack categories, show that all categories have excellent accuracy, with U2R attacks having the highest accuracy rate (99.83%). However, the F1-score varies greatly, indicating that it can be difficult to distinguish between R2L and U2R attacks.

The 3-layer DDQN model achieves 95.66% accuracy and 96.13% F1-score, greatly outperforming the 2-layer model in the AWID dataset. The breakdown of attack types reveals that regular and injection attacks have good precision and recall, whereas impersonation assaults pose a bigger challenge, as seen by their lower precision of 56.86%. The DDQN model effectively distinguishes between different types of attacks and exhibits robustness in addressing the imbalanced nature of the dataset.

The CSE-CIC-IDS dataset exhibits remarkable performance when DDQN is applied; both 2-layer and 3-layer models obtain F1-scores and nearly flawless accuracy. In general consistency, the 2-layer model performs somewhat better than the 3-layer model, while in precision; the 3-layer model is superior. With both obtaining above 99% in accuracy and recall, the evaluation metrics for benign and DDoS attacks highlight the model's excellent capabilities in successfully recognizing various kinds of cyber threats.

The training process is depicted in figures where the defense agents' total rewards increase throughout training sessions. This recommends that their exhibition in limiting attacks has moved along. Simultaneously, aggressor rewards definitely drop, exhibiting how well the model adjusts and learns progressively.

As per the review, DDQN models are quite great at distinguishing and ruining various types of attacks on an assortment of datasets. The commitment of these models to further develop cyber security defenses is featured by their capacity to actually arrange different sorts of attacks and their vigour in overseeing imbalanced datasets. It is proposed from the outcomes that much less difficult models, like the 2-layer DDQN, can accomplish huge execution, making them valuable for real applications, even while deeper network designs can offer slight increments. The imaginative utilization of DDQN in cyber security opens the entryway for assurance components that are more versatile and strong to changing cyber threats.

6. Conclusion And Future Work

We present the DDQN, a reinforcement learning model for ordering and recognizing various kinds of network cyber attacks. Three datasets are utilized to approve the model, and DQNs are utilized to make a DRL approach. Reinforcement learning is associated with the DNN to permit it to interface with the network environment. The independent way of behaving of DDQN specialists is utilized to accumulate and examine network traffic to recognize hurtful network payloads. Besides, DDQN and other DRL techniques have the advantage of extraordinarily diminishing expectation times, which makes them ideal for online location and the demands of contemporary network administrations. Besides, to further develop learning abilities, we direct an extensive examination of various DDQN specialist boundaries, including the markdown factor, bunch size, and number of learning episodes, to decide the best tweaking methodologies for network cyber assault identification undertakings. Our trial results show that the proposed DDQN model has a viable learning limit since it can precisely and freely arrange different sorts of network dangers. The DDQN model's predominance is featured by examination with before research, featuring its true capacity for powerful cyber protective strategies. In light of everything, this study underlines that it is so vital to utilize state of the art computer based intelligence strategies to counter different web-based risks and further develop cyber security procedures. We mean to try our recommended approach in a genuine cloud environment as a feature of our next work. Through this arrangement, the DDQN specialist will actually want to work on its ability for self-learning and precisely group dangers in situations that happen continuously. Likewise, we need to assess the generalizability and helpfulness of the DDQN model by utilizing it to recognize ransom ware.

References

[1.] Alavizadeh, H., Alavizadeh, H., & Jang-Jaccard, J. (2022). Deep Q-learning based reinforcement learning approach for network intrusion detection. Computers, 11(3), 41.

[2.] Basnet, R. B., Shash, R., Johnson, C., Walgren, L., & Doleck, T. (2019). Towards detecting and classifying network intrusion traffic using deep learning frameworks. Journal of Internet Services and Information Security, 9(4), 1–7.

[3.] Bhattacharya, A., Ramachandran, T., Banik, S., Dowling, C. P., & Bopardikar, S. D. (2020). Automated adversary emulation for cyber-physical systems via reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 1–6). IEEE.

[4.] Praveen, S. P., Sindhura, S., Srinivasu, P. N., & Ahmed, S. (2023, September). Combining CNNs and Bi-LSTMs for Enhanced Network Intrusion Detection: A Deep Learning Approach. In *2023 3rd International Conference on Computing and Information Technology (ICCIT)* (pp. 261-268). IEEE.

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

[5.] Mahmoud M. Ismail, Ahmed A. Metwaly. "Enhancing Wireless Ad-Hoc Network Security by Mitigating Distributed Denial-of-Service (DDoS) Attacks." Full Length Article, Vol. 8, No. 2, 2024 ,PP. 46-52 (Doi : https://doi.org/10.54216/IJWAC.080205)

[6.] Dong, S., Xia, Y., & Peng, T. (2021). Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. IEEE Transactions on Network and Service Management, 18(4), 4197–4212.

[7.] Dutta, A., Chatterjee, S., Bhattacharya, A., & Halappanavar, M. (2023). Deep reinforcement learning for cyber system defense under dynamic adversarial uncertainties. arXiv preprint arXiv:2302.01595.

[8.] Franco, M. F., Sula, E., Huertas, A., Scheid, E. J., Granville, L. Z., & Stiller, B. (2022). SecRiskAI: A machine learning-based approach for cybersecurity risk prediction in businesses. In Proceedings of the 2022 IEEE 24th Conference on Business Informatics (CBI) (Vol. 1, pp. 1–10). IEEE.

[9.] Haque, N. I., Shahriar, M. H., Dastgir, M. G., Debnath, A., Parvez, I., Sarwat, A., & Rahman, M. A. (2020). Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey. arXiv preprint arXiv:2010.00661.

[10.] Huang, Y., Huang, L., & Zhu, Q. (2022). Reinforcement learning for feedback-enabled cyber resilience. Annual Review of Control, 53, 273–295.

[11.] Khaw, Y. M., Jahromi, A. A., Arani, M. F., Sanner, S., Kundur, D., & Kassouf, M. (2020). A deep learning-based cyberattack detection system for transmission protective relays. IEEE Transactions on Smart Grid, 12(3), 2554-2565.

[12.] Landen, M., Chung, K., Ike, M., Mackay, S., Watson, J. P., & Lee, W. (2022). DRAGON: Deep reinforcement learning for autonomous grid operation and attack detection. In Proceedings of the 38th Annual Computer Security Applications Conference (pp. 13–27).

[13.] Bikku, T., Chandolu, S. B., Praveen, S. P., Tirumalasetti, N. R., Swathi, K., & Sirisha, U. (2024). Enhancing Real-Time Malware Analysis with Quantum Neural Networks. *Journal of Intelligent Systems and Internet of Things*, *12*(1), 57-7.

[14.] Meier, R., Lavrenovs, A., Heinäaro, K., Gambazzi, L., & Lenders, V. (2021). Towards an AI-powered player in cyber defence exercises. In Proceedings of the 2021 13th International Conference on Cyber Conflict (CyCon) (pp. 309–326). IEEE.

[15.] Mahmoud A. Zaher, Mohmaed A. Labib, Artificial Flora Optimization Algorithm with Functional Link Neural Network for DoS Attack Classification in WSN, Journal of International Journal of Wireless and Ad Hoc Communication, Vol. 4 , No. 1 , (2022) : 08-18 (Doi : https://doi.org/10.54216/IJWAC.040101)

[16.] Mesadieu, F., Torre, D., & Chennameneni, A. (2024). Leveraging deep reinforcement learning technique for intrusion detection in SCADA infrastructure. IEEE Access.

[17.] Reddy, A. S., Praveen, S. P., Ramudu, G. B., Anish, A. B., Mahadev, A., & Swapna, D. (2023, January). A network monitoring model based on convolutional neural networks for unbalanced network activity. In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1267-1274). IEEE.

[18.] Paidipati, K. K., Kurangi, C., Uthayakumar, J., Padmanayaki, S., Pradeepa, D., & Nithinsha, S. (2024). Ensemble of deep reinforcement learning with optimization model for DDoS attack detection and classification in cloud based software defined networks. Multimedia Tools and Applications, 83(11), 32367-32385.

[19.] Piplai, A., Anoruo, M., Fasaye, K., Joshi, A., Finin, T., & Ridley, A. (2022). Knowledge guided Two-player Reinforcement Learning for Cyber Attacks and Defenses. In Proceedings of the International Conference on Machine Learning and Applications, Nassau, Bahamas.

[20.] Radoglou-Grammatikis, P., Rompolos, K., Sarigiannidis, P., Argyriou, V., Lagkas, T., Sarigiannidis, A., ... & Wan, S. (2021). Modeling, detecting, and mitigating threats against industrial healthcare systems: A combined software defined networking and reinforcement learning approach. IEEE Transactions on Industrial Informatics, 18(3), 2041-2052.

[21.] Randhawa, R. H., Aslam, N., Alauthman, M., Khalid, M., & Rafiq, H. (2023). Deep reinforcement learning based evasion generative adversarial network for botnet detection. SSRN.

[22.] Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model. Scientific Reports, 12(1), 15370.

[23.] Salam, A., Ullah, F., Amin, F., & Abrar, M. (2023). Deep learning techniques for web-based attack detection in industry 5.0: A novel approach. Technologies, 11(4), 107.

[24.] Sarker, I. H. (2022). Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview. Security and Privacy, 6, e295.

[25.] Selim, A., Zhao, J., Ding, F., Miao, F., & Park, S. Y. (2023). Adaptive deep reinforcement learning algorithm for distribution system cyber attack defense with high penetration of DERs. IEEE Transactions on Smart Grid.

[26.] Mahmoud A. Zaher, Nabil M. Eldakhly, Cyber Attack Detection in Wireless Adhoc Network using Artificial Intelligence, Journal of International Journal of Wireless and Ad Hoc Communication, Vol. 6 , No. 2 , (2023) : 18-33 (Doi : https://doi.org/10.54216/IJWAC.060202).

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024

[27.] Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., Alharbi, F., & Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security Applications, 72, 103405.

[28.] Tharewal, S., Ashfaque, M. W., Banu, S. S., Uma, P., Hassen, S. M., & Shabaz, M. (2022). Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. Wireless Communications and Mobile Computing, 2022, 1-8.

[29.] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550.

[30.] Wu, C., Pan, W., Staa, R., Liu, J., Sun, G., & Wu, L. (2023). Deep reinforcement learning control approach to mitigating actuator attacks. Automatica, 152, 110999.

[31.] S. Phani Praveen , Thulasi Bikku, P. Muthukumar, K. Sandeep, Jampani Chandra Sekhar, V. Krishna Pratap. (2024). Enhanced Intrusion Detection Using Stacked FT-Transformer Architecture. Journal of , 13 (2), 19-29 (**Doi :** https://doi.org/10.54216/JCIM.130202)

[32.] Biyyapu, N., Veerapaneni, E. J., Surapaneni, P. P., Vellela, S. S., & Vatambeti, R. (2024). Designing a modified feature aggregation model with hybrid sampling techniques for network intrusion detection. Cluster Computing, 1-19.

[33.] Aruna, R., Kushwah, V. S., Praveen, S. P., Pradhan, R., Chinchawade, A. J., Asaad, R. R., & Kumar, R. L. (2024). Coalescing novel QoS routing with fault tolerance for improving QoS parameters in wireless Ad-Hoc network using craft protocol. Wireless Networks, 30(2), 711-735.

[34.] Jyothi, V. E., Kumar, D. L. S., Thati, B., Tondepu, Y., Pratap, V. K., & Praveen, S. P. (2022, December). Secure data access management for cyber threats using artificial intelligence. In 2022 6th International Conference on Electronics, Communication and Aerospace Technology (pp. 693-697). IEEE.

Doi : https://doi.org/10.54216/JCIM.140107
Received: January 22, 2024 Revised: March 19, 2024 Accepted: May 26, 2024