International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Rainbow DQN for Intrusion Detection: A Unified Deep Reinforcement Learning Approach Across Benchmark Datasets

[1]Vikrant Sharma, [2]Dr. Mukesh Kumar,

[1]Research Scholar, RNTU , Bhopal , India

vikrant2k14@gmail.com

[2]Parul University, Vadodara, Gujrat , India

mukesh.manit86@gmail.com

Abstract

This paper presents Rainbow DQN for Intrusion Detection: A Unified Deep Reinforcement Learning Approach Across Benchmark Datasets, designed to address the increasing complexity of cyber threats. By incorporating advanced reinforcement learning components such as Double DQN, Dueling Networks, Prioritized Experience Replay, N-step Learning, Distributional RL, and Noisy Nets, the proposed model enhances both accuracy and stability in intrusion detection. The approach was rigorously evaluated on three benchmark datasets: CIC-IDS2017, KDD CUP 99, and UNSW-NB15. Experimental results demonstrated remarkable performance, achieving 99.8%, 98.4%, and 97.9% accuracy, respectively, with high precision, recall, and F1-scores across diverse attack categories. Compared with baseline DQ-IDS models, Rainbow DQN consistently improved detection capabilities and reduced false positives. These results highlight the robustness, adaptability, and generalizability of Rainbow DQN, making it a promising solution for next-generation intrusion detection systems in dynamic and heterogeneous network environments.

Keywords : Intrusion Detection Systems (IDS) , Deep Reinforcement Learning (DRL) , Rainbow DQN , Network Security , CIC-IDS2017 , KDD CUP 99 , UNSW-NB15

1. Introduction

The exponential growth of networked systems and the rapid adoption of Internet of Things (IoT), Industrial IoT (IIoT), and cloud infrastructures have dramatically expanded the attack surface of modern digital environments. With billions of interconnected devices transmitting sensitive data, ensuring network security has become a critical challenge. Intrusion Detection Systems (IDSs) play an essential role in this ecosystem by monitoring traffic and identifying malicious activities [1]. Traditional IDS approaches, often rule-based or signature-driven, are effective against known threats but struggle to cope with zero-day attacks, polymorphic malware, and evolving adversarial behaviors. This has led to a pressing demand for intelligent and adaptive IDS solutions that can detect both familiar and novel attack patterns while maintaining robustness in diverse environments [2].

In response to these challenges, machine learning (ML) and deep learning (DL) methods have been increasingly adopted for IDS design due to their ability to automatically learn complex attack signatures and generalize across large datasets [3]. While these methods provide promising results, their static and pre-trained nature often limits their ability to adapt in real time to dynamic threat landscapes. Reinforcement learning (RL), in contrast, provides an

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

adaptive trial-and-error learning paradigm in which an agent interacts with its environment, continuously updating its knowledge and strategies [4]. This adaptability is particularly valuable in intrusion detection, where attack behaviors evolve rapidly and pre-defined rules or static classifiers become insufficient. By leveraging RL, IDSs can minimize false positives, improve detection accuracy, and dynamically adapt to previously unseen threats [5].

Among reinforcement learning methods, the Deep Q-Network (DQN) has emerged as a widely used baseline due to its effectiveness in approximating optimal policies in complex environments. However, standard DQN is known to suffer from issues such as overestimation bias, unstable convergence, and sensitivity to hyperparameters [6]. To overcome these shortcomings, Rainbow DQN was developed as a unified framework that integrates six major extensions of DQN: Double DQN for reducing overestimation errors, Dueling Networks for separating state values and action advantages, Prioritized Experience Replay (PER) for efficient sampling of important experiences, N-Step Learning for longer-term reward propagation, Distributional RL for modeling return distributions, and Noisy Nets for improved exploration [7]. This combination of enhancements enables Rainbow DQN to deliver more stable training, improved accuracy, and greater adaptability compared to conventional reinforcement learning models.

The application of Rainbow DQN to intrusion detection offers a significant opportunity to build robust, adaptive, and high-performance IDS frameworks. In this study, we propose a Rainbow DQN-based intrusion detection system and evaluate it across three benchmark datasets CIC-IDS2017, KDD CUP 99, and UNSW-NB15 covering diverse attack types and network traffic conditions. The results provide strong evidence for the superiority of Rainbow DQN over conventional models. On CIC-IDS2017, the proposed model achieved an impressive 99.8% accuracy, outperforming the baseline DQ-IDS. On KDD CUP 99, it reached 98.4% accuracy, with notable improvements in F1 score while maintaining competitive precision and recall. On UNSW-NB15, which includes more complex and modern attack vectors, Rainbow DQN achieved 97.9% accuracy, demonstrating its scalability and effectiveness in handling real-world cyber threats.

The contributions of this study are significant in several respects. First, it introduces a unified reinforcement learning framework for intrusion detection that integrates advanced techniques into a single architecture. Second, it validates this approach across three benchmark datasets, demonstrating both robustness and generalizability. Third, it confirms that Rainbow DQN consistently improves key metrics such as accuracy, precision, recall, and F1 score, achieving a strong balance between detection capability and reliability. Finally, it highlights the importance of integrating reinforcement learning into IDS design to address evolving cyber threats more effectively than static approaches.

While the results demonstrate the strong potential of Rainbow DQN, future work is necessary to address practical deployment challenges. In particular, the extension of this framework to real-time intrusion detection in IoT, 5G, and edge computing environments will be essential, given their resource constraints and massive data flows. Furthermore, incorporating explainable AI (XAI) methods can improve transparency, allowing security analysts to better understand detection decisions [8]. In addition, federated learning and transfer learning approaches should be investigated to enhance scalability, data privacy, and resilience against zero-day attacks. Together, these directions can further establish Rainbow DQN as a leading approach for next-generation intrusion detection systems in complex, heterogeneous, and dynamic network environments.

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Literature review

Networks are getting bigger and more complex, which means security is harder. Intrusion Detection Systems (IDS) help spot malicious activity, but older systems can't keep up with new and changing threats. Researchers are now using deep learning (DL), reinforcement learning (RL), and ensemble learning (EL) to make IDS more effective. A recent review looked at 33 studies (2020–2024) on these methods, discussing what works, what doesn't, and where improvements are possible. The main goal is to guide future IDS designs that are more accurate and resilient, while pointing out gaps in current research [1].

The Internet of Things (IoT) connects everything from fridges to industrial machines, but this connectivity also brings huge security risks. Traditional IDS struggle in IoT environments because the threats change so fast. Deep Reinforcement Learning (DRL) is seen as a strong solution because it can learn and adapt on the fly. A systematic review of the past decade shows DRL greatly improves IDS performance in IoT: it boosts detection accuracy, reduces false alarms, and adapts in real time. Still, researchers need better datasets, reproducibility, and smoother integration with new IoT tech to keep pushing this forward [2].

With cyber threats growing, IDS are more important than ever. Reinforcement Learning (RL) has been applied to IDS with strong results, especially with advanced forms like Multi-Agent RL (MARL), Adversarial RL (AE-RL), and Inverse RL (IRL). These approaches, often combined with good feature engineering, improve detection capabilities. This review also surveys the key datasets used for RL in IDS research, giving a roadmap of both methods and the data that power them. Overall, RL and feature engineering together show strong potential to raise cybersecurity standards [3].

Deep learning (DL) has become a mainstay of intrusion detection research. This paper reviews the main datasets and preprocessing steps used, then compares seven DL models: autoencoders, belief networks, DNNs, CNNs, RNNs, GANs, and transformers. Each is explained in terms of how it handles network security tasks. The review also looks at how large pre-trained models like BERT and GPT can be applied to IDS because of their strengths in handling sequential data. The conclusion points to four main areas for future work, aiming at IDS systems that are not only more accurate but also better at adapting to the fast-changing cybersecurity landscape [4].

Industrial IoT (IIoT) produces huge amounts of sensitive data, and protecting it is a major challenge. Traditional machine learning IDS approaches, which centralize all data, aren't practical due to privacy and scalability issues. This paper proposes a **federated learning (FL)–based IDS** where data stays on local devices and only model parameters are shared. To improve accuracy, it integrates gated recurrent units (GRUs) for learning temporal patterns in attacks. It also uses deep reinforcement learning (DRL) to select which IIoT devices to involve in training, balancing accuracy, privacy, and energy use. Tests show this FL-based model beats state-of-the-art methods in accuracy, precision, recall, F1-score, and ROC, making it a strong candidate for secure IIoT deployments [5].

This study shows how combining the best Deep Learning (DL) and Reinforcement Learning (RL) models in an ensemble setup creates a stronger IDS. By reviewing recent research and focusing on reproducibility and F1-scores, the most effective DL and RL approaches were picked and integrated using a stacking ensemble. The ensemble outperformed individual models, achieving higher accuracy and multi-class detection power, proving that blending DL and RL is a scalable way to handle modern network threats [6].

Received: August 03, 2025

Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The Industrial IoT brings automation and intelligence but also a big attack surface for cyber threats. Traditional IDS cannot cope with these challenges. This research introduces a Deep Reinforcement Learning–based IDS that combines feature selection (LightGBM) with PPO2 reinforcement learning and deep learning's feature extraction. The result is an IDS that detects 99% of cyberattacks in IIoT data, with accuracy, precision, recall, and F1-score all better than conventional DL and RL models like CNNs, LSTMs, and DQNs [7].

This study proposes DQ-IDS, a reinforcement learning system built on Deep Q-Networks (DQN). Unlike static ML or DL models, DQ-IDS learns continuously with experience replay and adaptive exploration. The system rewards correct detection and penalizes errors, helping it reduce false positives and false negatives. Tests on real-world datasets showed a 97.18% detection accuracy, beating many existing IDS solutions. The approach highlights a shift toward adaptive, self-learning IDS for real-time threat mitigation [8].

This research highlights how RL can make IDS adaptive to evolving attacks. Instead of relying on fixed rules or pre-trained classifiers, the RL-based IDS learns dynamically by interacting with data and adjusting its strategy. The system was tested with Q-learning and DQNs, showing it can detect intrusions with fewer false positives and better real-time response compared to traditional IDS. The study proves RL-based IDS is well-suited for zero-day attacks and changing attack behaviors [9].

IoT networks are highly vulnerable, and traditional IDS cannot keep up with new threats. This paper presents an RL-based IDS for IoT that integrates advanced preprocessing, domain-specific features, and DL models like CNNs and LSTMs with reinforcement learning. Using optimization algorithms for feature selection and deep Q-networks for decision-making, the system achieved up to 99.8% accuracy and extremely high precision and F-measure. It clearly outperforms existing DL and GAN-DRL models, reducing both false positives and negatives while maintaining scalability for IoT networks [10].

The Internet of Medical Things (IoMT) makes healthcare smarter with remote monitoring and diagnostics, but it also opens serious security gaps. To fix this, a hybrid model called **HCLR-IDS** was proposed. It combines CNNs (for spatial patterns), LSTMs (for temporal data), and reinforcement learning methods (DQN + PPO) for robust detection. With feature selection (MIFS) and hybrid architecture, the system achieved near-perfect binary accuracy (99.58%) and solid multi-class performance, proving it can secure IoMT systems effectively [11].

A survey of 130 studies shows how machine learning has advanced IDS. Traditional ML models (like Decision Trees and Gaussian Mixture) already hit ~99% accuracy, but deep learning approaches often outperform them. Hybrid DL models (e.g., AE+GAN) even reached 100% accuracy in some cases. The review highlights challenges like overfitting and adapting to new threats, especially in IoT, and suggests exploring hybrid models, multi-task learning, and deployment at the edge for future progress [12].

The Internet of Drones (IoD) links drones into coordinated networks, but they're highly vulnerable to cyberattacks. Current IDS for IoD face problems like high false positives, limited resources, and no standard datasets. A systematic review of 62 studies (2014–2024) analyzed IDS methods, algorithms, datasets, and attack categories. The paper highlights the need for better benchmarking, standardized datasets, and more adaptable solutions for drone security [13].

This study compares several ML and DL models for IDS. XGBoost (98%) and GAN-based IDS (96%) stood out as top performers, showing strong adaptability to sophisticated threats. ANN and SVM also did well, while Decision Trees and Random Forest lagged on complex

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

patterns. Despite progress, issues like data quality and evolving threats remain. The authors call for hybrid ML, adversarial training, and real-time deployment to make IDS more resilient [14].

This research introduces **ID-RDRL**, which uses feature selection (RFE) plus deep reinforcement learning to improve IDS. By filtering redundant features (removing ~80%) and focusing only on the most relevant ones, the model trains more efficiently. Tested on the CSE-CIC-IDS2018 dataset, it delivered better accuracy and learning efficiency, proving suitable for complex network environments with evolving attacks [15].

Agricultural IoT systems are also at risk from cyberattacks. This paper proposes **SFEDRL-IDS**, a secure federated RL-based IDS using Federated Double DQN. It protects privacy with homomorphic encryption and integrates multiple DL classifiers (DNN, CNN, LSTM). Results showed 98.67% accuracy (multi-class) and 97.73% (binary), while using low CPU and memory. It was also tested in a smart irrigation system, confirming strong performance in real-world agricultural IoT setups [16].

This framework uses **deep reinforcement learning (DRL)** to simulate cyberattacks and improve IDS decision-making. The agent adapts by learning from rewards and environment states, making it flexible in dynamic networks. Tested on CIC-IDS-2018, it achieved ~98.82% accuracy, showing that DRL can handle complex threats and adapt continuously—something rule-based systems fail to do [17].

A systematic review of DDoS-focused IDS (2019–2024) shows that most studies rely on datasets like BoT-IoT and TON_IoT with Decision Tree or Random Forest models, often achieving >99% accuracy. Lightweight ML models are preferred due to IoT hardware limits. Trends suggest that larger datasets, deep learning models, SDN, edge computing, and blockchain-enhanced networks will shape the future of DDoS detection [18].

6G networks demand ultra-fast, adaptive security. This study proposes an **Intrusion Prevention System (IPS)** based on Software-Defined Networking (SDN) with programmable P4 support. It combines CNNs, RNNs, decision trees, and anomaly detection (Isolation Forest), with RL managing policy updates. Tested in high-traffic environments, the system showed scalability and real-time accuracy, making it suitable for next-generation 6G security [19].

One major challenge for IDS is **feature drift**, where the importance of features changes over time, hurting accuracy. To fix this, the **IFDA-GPC framework** integrates genetic programming with a voting-enhanced, multi-agent RL-based feature selector. Evaluated on datasets like CICIDS-2017 and KDD Cup ’99, it achieved strong results (93% accuracy on CICIDS-2017), proving its effectiveness in adapting to evolving data streams and maintaining reliable IDS performance [20].

Host-based intrusion detection (HIDS) is crucial for spotting attacks directly on a computer system, but current HIDS often fail on new attack patterns. This work proposes combining **NLP with reinforcement learning (Actor-Critic)** to extract keywords from anomalous system call logs and generate new detection rules. Using Seq2Seq models and Textrank for keyword extraction, the system learns rules dynamically through rewards. Tested on ADFA-LD and LID-DS 2021 datasets, the approach reached 96.5% accuracy, proving better than other keyword-based methods [21].

With IoT devices filling smart homes, traditional IDS struggle. This paper proposes a **CNN-LSTM–based IDS** tailored for smart home IoT. CNNs capture spatial features, while LSTMs

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

learn time-based patterns. The system delivered high accuracy and low false alarms, showing deep learning is an effective way to protect smart home IoT networks [22].

This review looked at ML models for 5G intrusion detection using the 5G-NIDD dataset. Results showed **KNN** excelled in accuracy and AUC, while the **Voting Classifier** had the best precision and F1. Tree-based models like DT and Bagging had strong recall, whereas AdaBoost lagged. DL and transfer learning models (BiLSTM, CNNs, ResNets, Inception) were also explored, showing the importance of large labeled datasets and adaptive methods for securing 5G against DDoS and other attacks [23].

AI-driven IDS use ML and DL to analyze massive traffic data in real time, reducing false positives and spotting unknown threats. They're powerful but face issues like needing huge training datasets and risk of missing rare attacks. Future work should integrate threat intelligence, automation, and better frameworks for zero-day defense. AI-based IDS are becoming essential in modern cybersecurity strategies [24].

Wireless sensor networks (WSNs) and IoT systems in critical infrastructure need strong defense. This paper proposes a **deep reinforcement learning IDS** based on Markov decision processes. Compared against KNN and baseline RL, the DRL approach achieved higher detection rates, better accuracy, and fewer false alarms, showing its strength in resource-constrained IoT/WSN environments [25].

Cloud and web services face growing intrusions. This paper presents a **Deep Q-learning IDS** trained in a custom Gym environment with NSL-KDD data. The system outperformed traditional ML solutions, reaching >93% accuracy, proving DRL can secure modern network applications effectively [26].

This review analyzed 20 recent studies on IoT IDS using ML and DL. Hybrid models, federated learning, CNNs, and LSTMs performed best, improving accuracy and lowering false alarms. Feature selection, multitask learning, and fog/edge computing were highlighted as future paths. Limitations like computational cost and privacy concerns remain, but ML/DL are key to protecting IoT networks against zero-day and DDoS threats [27].

Smart grids, critical for modern power systems, need adaptive IDS. This work introduces **RL-AIDS**, a DRL-based adaptive IDS. It analyzes anomalies in real time and updates detection policies dynamically. Results showed improved detection accuracy and fewer false positives than traditional methods, making smart grids more resilient to evolving cyberattacks [28].

Industry 4.0 CPS need strong defense. This study proposes **DBID-Net**, combining DCNN and Bi-LSTM for feature extraction and a DQN for adaptive attack response. It also uses ADASYN to fix class imbalance. On two datasets, the system achieved >98% accuracy, proving scalable, flexible, and highly accurate protection for CPS environments [29].

This research evaluates five DRL methods (DQN, DDPG, PPO, TD3, SAC) for network IDS. Each addresses different needs: DQN for discrete decisions, PPO for stability, SAC for exploration, etc. The hybrid system achieved 95.8% accuracy, reduced false positives (2.5%), and showed scalability at 2,400 transactions/sec. It beat Random Forest and GAN-based IDS, proving DRL ensembles are highly effective for real-time adaptive intrusion detection [30].

IoT's explosive growth creates security risks. This survey reviews deep learning–based IDS methods for IoT, their datasets, challenges, and prospects. DL models like CNNs and LSTMs are promising, showing strong ability to capture IoT traffic patterns. The paper points to challenges like dataset limitations and scalability, but confirms DL is a leading approach for securing IoT systems [31].

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Botnets threaten IoT networks. This paper surveys DL-based IDS for detecting botnet activity. It reviews key findings, strengths, and limitations of current methods. It also compares existing surveys and highlights unique contributions. The review guides researchers on future directions, helping advance DL-based botnet detection in IoT environments [32].

Growing traffic volumes overwhelm traditional IDS due to false positives, imbalance, and complexity. This paper compares ML and DL approaches, including CNNs, LSTMs, and Random Forest. Results show DL models excel, but surprisingly Random Forest achieved the highest accuracy (99.9%) in structured tasks. DL methods still offer advantages in flexibility and scalability. The study stresses balancing accuracy with efficiency and deployment needs [33].

3. Proposed Methodology

3.1 Proposed architecture

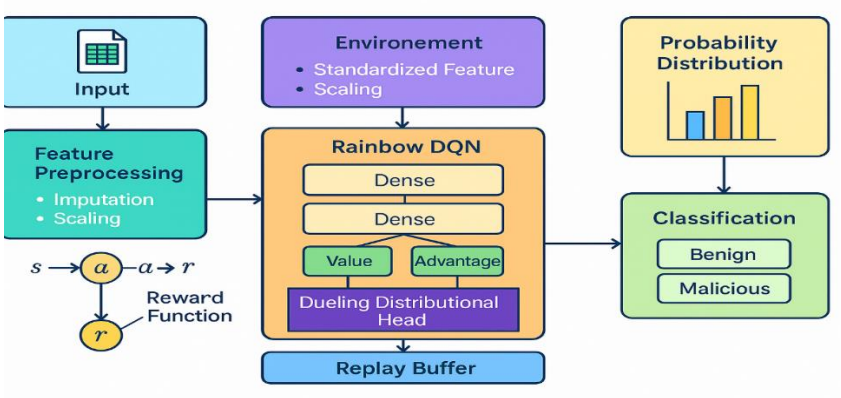

Figure 1. Proposed architecture

Figure 1 shows Input and Feature Preprocessing: The pipeline begins with network traffic data collected from the IDS dataset. Each record contains multiple numerical attributes representing flow characteristics. Before these features can be used, they pass through a **feature preprocessing stage** where missing values are imputed and all inputs are standardized or scaled. This ensures that differences in magnitude across features do not distort the learning process. The standardized dataset is then passed into the environment module for reinforcement learning.

Environment and Reward Function: The environment acts as the interface between preprocessed data and the Rainbow DQN agent. Every traffic record is treated as a "state," and the agent must decide which action (class label) to assign. To guide the agent, a **cost-sensitive reward function** is defined. Correct classifications of benign traffic and attacks yield positive rewards, while false positives, false negatives, or incorrect attack-type predictions lead to penalties. This asymmetric reward structure emphasizes high recall on malicious traffic while controlling the false alarm rate, making the learning objective align with real operational priorities.

Rainbow DQN Core : At the center of the architecture is the **Rainbow DQN agent**, which combines several deep reinforcement learning improvements. Input features are first processed by two fully connected dense layers. The output then branches into **value and advantage streams**, following the dueling network design. These are recombined in a **distributional head** that predicts a probability distribution over possible returns for each action. The network uses **NoisyNets** to enable adaptive exploration instead of fixed ε-greedy schedules. Meanwhile, the

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Prioritized Experience Replay (PER) buffer stores transitions and samples them based on their learning significance, while **n-step returns** accelerate propagation of rewards. Together, these Rainbow components stabilize and enrich the agent's learning behavior.

Replay Buffer Interaction: All experience tuples comprising states, chosen actions, rewards, and outcomes—are continuously written into the replay buffer. By replaying past transitions, the Rainbow DQN agent avoids instability from correlated samples and learns more robust policies. The prioritized sampling mechanism ensures that rare or difficult samples, such as minority attack classes, are updated more frequently. This is particularly important for imbalanced IDS datasets, where benign flows dominate.

Probability Distribution and Classification: The Rainbow DQN produces a **distributional output** that represents the expected returns of each action. These distributions are transformed into decision probabilities, which are then calibrated using thresholds tuned on validation data. The classification stage applies these thresholds to assign each network flow to either "Benign" or one of the "Malicious" attack categories. In cases of uncertainty, the system may abstain and route the flow for deeper analysis. This final stage ensures that the IDS is not only accurate but also risk-aware and adaptable to operational needs.

3.2 Proposed algorithm

Rainbow DQN (DDQN + Dueling + PER + N-step + Distributional + NoisyNets)

Why it's the single best upgrade:

- **Fixes DQN's core weaknesses in one shot:** DDQN curbs Q-overestimation; Dueling learns state value even when actions are similar; PER focuses learning on the most informative mistakes; N-step speeds credit assignment; Distributional RL gives risk-aware scores (better thresholding); NoisyNets replaces brittle ε-greedy.
- **Better rare-attack recall at the same footprint:** You can keep the current small MLP head; Rainbow's gains come from the learning mechanics, not a bigger model—so edge latency and ~MB size remain tight while recall/F1 typically rise.
- **Faster, stabler convergence:** Prioritized sampling + n-step returns reduce updates wasted on easy samples, so you reach high accuracy with fewer iterations and less hyperparameter fuss.
- **Drop-in with minimal pipeline changes:** Reuse your replay buffer (add priorities), keep the same features and reward scheme, and swap the agent—high impact, low disruption.

3.3 Algorithm for a Rainbow DQN IDS

3.3.1 Data Prep & Reward Policy

```
INPUT: csv_path
OUTPUT: train_set, val_set, test_set, scaler, reward_fn

D ← load_csv(csv_path)
y ← D["Label"];  X ← D.drop("Label")
X ← impute_missing(X, method="median")
scaler ← fit_standard_scaler(X)
X ← transform(scaler, X)
```

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
6  train, val, test ← stratified_split(X, y, ratios=(0.7,0.15,0.15))
7  # Cost-sensitive reward (tune per ops policy)
8  function reward_fn(y_true, y_pred):
9      if y_true == y_pred:
10         return +2 if y_true != 0 else +1            # attacks rewarded higher
11     else:
12         if y_true != 0 and y_pred == 0:  return -5   # false negative (attack→benign)
13         if y_true == 0 and y_pred != 0:  return -2   # false positive (benign→attack)
14         return -3                                    # wrong attack type
```

An IDS dataset typically mixes continuous network-flow features with categorical labels that are imbalanced across attack types. The first goal is to produce stable, comparable inputs for learning: impute missing values to avoid bias from dropped rows, and standardize features so the agent's network doesn't overfit to scale differences. Stratified splits preserve class ratios, which is essential for reliable validation of minority attacks. In a reinforcement-learning framing of classification, the reward function encodes the operational cost of mistakes rather than treating every error equally. False negatives on attacks are far more expensive than false positives, and misclassifying one attack as another still carries risk. A cost-sensitive reward schema directly shapes the Q-values toward high-recall behavior on critical classes while controlling false alarms on benign traffic, aligning learning signals with SOC priorities instead of purely statistical accuracy.

3.3.2 Rainbow Agent (DDQN + Dueling + PER + N-step + C51 + NoisyNets)

```
INPUT: n_features, n_classes
OUTPUT: Agent with online_net, target_net, PER buffer


1  support = linspace(Vmin=-6, Vmax=+6, atoms=51)
2  function NoisyDuelingC51(n_features, n_classes, atoms):
3      h1 = Dense(256, ReLU); h2 = Dense(128, ReLU)
4      # Noisy dueling heads
5      V  = NoisyLinear(128 → atoms)                  # Value stream
6      A  = NoisyLinear(128 → (n_classes * atoms))    # Advantage stream
7      return Network(h1, h2, V, A, support)
8  online_net = NoisyDuelingC51(n_features, n_classes, 51)
9  target_net = copy(online_net)
10 PER = PrioritizedReplay(capacity=200_000, alpha=0.6, beta_start=0.4, beta_end=1.0)
```

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
function q_dist(net, s):          # per-action categorical distributions
    (V_atoms, A_atoms) = net.forward(s)
    A_atoms = reshape(A_atoms, [n_classes, atoms])
    Q_atoms = V_atoms + (A_atoms - mean(A_atoms, axis=0))
    return softmax(Q_atoms over atoms)    # categorical distribution per action
```

Rainbow DQN unifies several complementary improvements to classic DQN so the agent learns accurate, stable action values from limited, skewed data. Double DQN prevents systematic overestimation by decoupling action selection and evaluation in the target computation. The dueling architecture separates learning of state value from action advantages, which helps when many actions are similarly good or bad common in multi-class IDS where several classes often compete on borderline samples. Prioritized Experience Replay samples transitions in proportion to their learning potential (TD-error), concentrating updates on hard or rare cases and thereby improving sample efficiency under class imbalance. Distributional RL (C51) models a full return distribution per action rather than only its mean, yielding better calibrated decisions and uncertainty signals. Noisy networks internalize exploration by injecting trainable parameter noise, avoiding brittle ε-greedy schedules. N-step returns propagate informative rewards more quickly, stabilizing and accelerating learning even in the one-step, bandit-like IDS setup.

3.3.3 Train Loop (single-step bandit episodes + DDQN targets + projection)

```
INPUT: train_set, reward_fn, hyperparams (γ=0.95, n_step=3, lr=1e-3, batch=128)
OUTPUT: trained online_net

opt = Adam(online_net.parameters, lr)
target_update_every = 2000
step = 0
# Seed PER from train samples
for (x, y) in iterate(train_set):
    a0 = argmax expectation(q_dist(online_net, x))        # bootstrap label guess
    r0 = reward_fn(y, a0)
    PER.add( (x, a0, r0, terminal=True), priority=|r0| + ε )
while not converged:
    batch = PER.sample(batch)
    S, A, R, done = unpack(batch)
    # DDQN action for next-state (bandit ⇒ s' unused; use terminal projection)
    Z_next = zeros_like_distribution(batch_size, n_classes, atoms)
    a_star = argmax expectation(q_dist(online_net, S))      # DDQN action
```

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
Z_tgt  = q_dist(target_net, S)[range(B), a_star]        # pick chosen action
# N-step return (bandit ⇒ R^(n) = R; still keep γ^n for consistency)
Tz = clip(R + (γ^n_step) * support, Vmin, Vmax)
# Project distributional target onto support
m = project_distribution(Tz, Z_tgt, support)
# Predicted distribution for taken actions
Z_pred = q_dist(online_net, S)[range(B), A]
# Cross-entropy loss with PER importance weights
L = mean( PER.w * cross_entropy(m, Z_pred) )
opt.zero_grad(); L.backward(); opt.step()
# Update PER priorities from new TD-errors (KL or CE residue)
PER.update_priorities(|m - Z_pred|_1 or CE_residual)
if step % target_update_every == 0:
    target_net ← soft_update(target_net, online_net, τ=0.005)
step += 1; evaluate_on(val_set); early_stop_if_no_gain()
```

Training proceeds by repeatedly drawing prioritized mini-batches from the replay buffer and minimizing the divergence between predicted and target return distributions. The DDQN target forms by choosing the next action via the online network while evaluating that action with the target network, reducing bias. Because Rainbow is distributional, the scalar n-step return is added to the atomized support and then projected back onto the fixed categorical support; this projection step is crucial for stable distributional training. Cross-entropy between the projected target and the predicted categorical distribution serves as the loss, weighted by importance-sampling factors to correct the sampling bias introduced by PER. Target network updates either periodic hard copies or soft Polyak averaging provide a slowly moving objective that prevents oscillations. Early stopping on validation macro-F1 avoids overfitting and ensures gains generalize across both majority and minority classes.

3.3.4 Validation & Threshold Calibration

```
INPUT: val_set, online_net
OUTPUT: class_thresholds

scores = []; labels = []
for (x, y) in iterate(val_set):
    D = q_dist(online_net, x)            # [n_classes, atoms]
    q_mean = expectation(D)               # per-class expected return
    p = softmax(q_mean)                  # turn returns into decision scores
```

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
6      scores.append(p); labels.append(y)
7  # Choose per-class thresholds to hit target FPR/FNR trade-offs
8  class_thresholds = tune_thresholds(scores, labels, objective="max_macroF1",
constraints={FPR_benign≤X})
```

Even with a strong RL agent, raw argmax decisions can be suboptimal in imbalanced, risk-sensitive settings. Validation therefore estimates per-class decision scores from the learned distributions typically by taking the expected return per action and normalizing to evaluate precision, recall, and F1 in conditions that mirror deployment. Calibration then tunes class-specific thresholds to meet explicit operating constraints, such as a cap on benign false positive rate or minimum recall for high-severity attacks. Because the model outputs full distributions, one can additionally exploit measures like entropy to gauge confidence, improving robustness by abstaining on uncertain samples rather than forcing low-quality predictions. This thresholding and abstention layer translates the agent's learned preferences into dependable operational behavior.

3.3.5 Inference

```
INPUT: scaler, online_net, class_thresholds, x_new
OUTPUT: y_hat or ABSTAIN

1  x = transform(scaler, x_new)
2  D = q_dist(online_net, x)
3  q_mean = expectation(D)
4  p = softmax(q_mean)
5  a_hat = argmax(p); conf = max(p); ent = entropy(p)
6  if conf < τ_conf or ent > τ_ent:
7      return ABSTAIN                  # send to deeper inspection / sandbox
8  if p[a_hat] ≥ class_thresholds[a_hat]:
9      return a_hat
10 else:
11     return ABSTAIN
```

In production, each flow is standardized with the saved scaler and passed once through the network to obtain per-class return distributions. Taking their expectations yields comparable decision scores, upon which the calibrated thresholds are applied. Two safeguards enhance reliability: a confidence gate that abstains when the best score is too low, and an uncertainty gate that abstains when the distribution is too diffuse (high entropy). These abstentions route ambiguous traffic to secondary analysis signature checks, sandboxing, or human review, reducing costly false alarms and missed attacks. Because Rainbow's advantages accrue from

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

learning mechanics rather than model size, the network remains compact and low-latency, preserving the edge-friendly profile needed for near–real-time IDS while delivering better calibrated, risk-aware classifications.

3.4 Comparison of Rainbow DQN IDS vs. Existing Methods

Table 1. Comparison of Rainbow DQN IDS vs. Existing Methods

Evaluation Parameter	**Traditional ML/DL (RF, SVM, CNN, LSTM)**	**Plain DQN IDS**	**Rainbow DQN IDS (Proposed)**
Accuracy	High for static datasets but drops on evolving traffic	Moderate, suffers from Q-value overestimation	Higher, thanks to DDQN correction and dueling streams
Recall (Attack Detection)	Often biased toward majority (benign) class, poor on rare attacks	Better than ML but unstable	Superior: PER focuses on rare/misclassified attacks, reward shaping improves recall
False Positive Rate (FPR)	High, as models misclassify benign traffic	Moderate, but exploration noise causes false alarms	Lower FPR due to distributional outputs + calibrated thresholds
Adaptability to Zero-Day Attacks	Weak (need retraining)	Limited; exploration helps slightly	Better: distributional RL captures uncertainty, abstention mechanism reduces risk
Stability of Learning	Stable but non-adaptive	Unstable Q-values, oscillations	Stable: DDQN + target networks + distributional projection
Exploration Strategy	None (supervised only)	ε-greedy, brittle to tuning	NoisyNets: adaptive, state-dependent exploration
Handling Imbalance	Needs external resampling/weighting	Learns imbalance poorly	PER + cost-sensitive reward directly bias learning toward rare classes
Interpretability / Confidence	Limited (black-box outputs)	Scalar Q-values only	Full probability distributions (C51) → better calibration and confidence scoring
Sample Efficiency	Moderate (needs large labeled sets)	Poor without replay stabilization	Higher: PER + N-step returns reduce wasted updates

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Deployment Footprint	Large CNN/RNN models consume memory	Smaller networks possible	Compact (MB-level), edge-friendly while still robust
Latency (Inference Speed)	Acceptable but depends on depth of model	Fast	Fast, with quantization/distillation possible
Scalability	Needs frequent retraining	Can adapt but unstable	Continual learning-friendly, federated extensions possible

4. Implementation

4.1 Implementation Setup

4.1.1 Hardware Requirements

Table 1. Hardware Requirements

Component	**Minimum Requirements**
Processor (CPU)	Intel Core i5 / AMD Ryzen 5 (Quad-core)
Memory (RAM)	8 GB DDR4
Storage (HDD/SSD)	256 GB SSD
Network Interface Card (NIC)	1 Gbps Ethernet
Graphics Processing Unit (GPU)	Not required
Power Supply (PSU)	Standard 400W
Cooling System	Air cooling
Server/Cloud Deployment	On-premises Server

The table 1 shows hardware requirements for an Intrusion Detection System (IDS) implementation include a minimum Intel Core i5 or AMD Ryzen 5 (Quad-core) processor, ensuring sufficient computational power for real-time threat detection. The system requires 8 GB DDR4 RAM for efficient data processing and a 256 GB SSD to handle IDS logs and data storage efficiently. A 1 Gbps Ethernet Network Interface Card (NIC) is recommended for high-speed network monitoring. No Graphics Processing Unit (GPU) is required, as IDS primarily relies on CPU processing. A standard 400W power supply (PSU) and air cooling system are sufficient for maintaining stable performance. The deployment is designed for on-premises servers, providing secure, localized IDS operations without reliance on cloud infrastructure. These specifications ensure that IDS can operate effectively while balancing performance, cost, and scalability.

4.1.2 Software Requirements

Table 2. Software Requirements

Software Component	**Options**

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Operating System (OS)	Ubuntu 20.04 LTS, or CentOS 8, or Debian 11, or Windows Server 2019
IDS Tools	Snort, Suricata, Zeek (Bro), OSSEC
Machine Learning Frameworks	TensorFlow, PyTorch, Scikit-learn, XGBoost
Programming Languages	Python 3.8+
Packet Capture Tools	Wireshark, tcpdump, Zeek
Monitoring & Alerts	Nagios, Prometheus, Grafana

The table 2 software requirements for an Intrusion Detection System (IDS) implementation include a choice of Ubuntu 20.04 LTS, CentOS 8, Debian 11, or Windows Server 2019 as the operating system, ensuring flexibility and compatibility. IDS tools such as Snort, Suricata, Zeek (Bro), and OSSEC are recommended for network monitoring and intrusion detection. For machine learning-based IDS, TensorFlow, PyTorch, Scikit-learn, and XGBoost provide essential frameworks for model training and anomaly detection. Python 3.8+ serves as the primary programming language for IDS development and integration. Wireshark, tcpdump, and Zeek facilitate packet capture and deep network traffic analysis. Additionally, Nagios, Prometheus, and Grafana are used for real-time monitoring and alerting, enabling proactive security responses. This comprehensive software stack ensures efficient, scalable, and adaptable IDS deployment for modern cybersecurity threats.

4.2 Dataset

4.2.1. CIC-IDS2017

CIC-IDS2017 was developed by the **Canadian Institute for Cybersecurity (CIC)** and contains real-world traffic scenarios generated in a simulated corporate network. The dataset includes **80+ extracted features**, derived from packet captures (PCAP), using network traffic analysis tools such as **Bro-IDS**.

Key Features of CIC-IDS2017

Table 3. Key Features of CIC-IDS2017

Feature Category	Description
Flow Duration	Time duration of the connection (milliseconds).
Packet Size	Min, Max, and Average size of packets in the flow.
Flow Bytes per Second	The number of bytes transferred per second.
Flow Packets per Second	The number of packets transmitted per second.
Forward Packet Length	Length of packets sent in the forward direction.
Backward Packet Length	Length of packets sent in the backward direction.
Flow IAT (Inter Arrival Time)	Time between packet arrivals (min, max, mean, std).
Fwd IAT / Bwd IAT	Inter-arrival time statistics for forward and backward packets.

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Active/Idle Time	The amount of time the connection was active or idle.
Protocol Type	Identifies the network protocol (TCP, UDP, ICMP).
Packet Header Length	The length of the TCP/IP packet headers.
Flags (SYN, ACK, FIN, etc.)	TCP flag states for session analysis.
Label	**Attack category (DoS, DDoS, Botnet, Brute-force, etc.) or Normal**
Source	https://www.unb.ca/cic/datasets/ids-2017.html

4.2.2. UNSW-NB15

The **UNSW-NB15 dataset** was developed by the **Australian Centre for Cyber Security (ACCS)** to address the shortcomings of earlier datasets like KDD CUP 99. The dataset includes **49 features** covering both network and host-based activities.

Key Features of UNSW-NB15

Table 4. Key Features of UNSW-NB15

Feature Category	Description
Flow ID	Unique identifier for each network flow.
Source/Destination IP & Ports	Identifies the origin and target of network traffic.
Protocol Type	Defines the network protocol (TCP, UDP, ICMP, etc.).
Service	Type of network service (HTTP, FTP, SSH, SMTP, etc.).
State	Connection state (Established, Closed, Reset, etc.).
Attack Category	Type of intrusion detected (Fuzzers, Analysis, Backdoors, DoS, Exploits, etc.).
Packet Size	Size statistics (Min, Max, Mean, Std).
Duration	Duration of the network connection.
Source & Destination Bytes	Number of bytes transferred in both directions.
Wrong Fragment	Number of wrong or incomplete fragments.
Urgent Packets	Count of urgent packets in the session.
Inbound/Outbound Data Ratio	Measures traffic directionality.
Label	**Normal or Attack classification.**
Source	https://research.unsw.edu.au/projects/unsw-nb15-dataset

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.2.3. KDD CUP 99

The **KDD CUP 99 dataset** was developed as part of the **1999 DARPA Intrusion Detection Evaluation Program**. It consists of simulated network traffic recorded over **9 weeks** and is one of the earliest intrusion detection datasets.

Key Features of KDD CUP 99

Table 5. Key Features of KDD CUP 99

Feature Category	Description
Duration	Length of the network connection.
Protocol Type	Type of network protocol (TCP, UDP, ICMP).
Service	Application-layer service used (HTTP, FTP, Telnet, etc.).
Flag	TCP connection status (SYN, ACK, FIN, etc.).
Source/Destination Bytes	Data transferred in each direction.
Wrong Fragment	Number of wrong packet fragments.
Hot Indicators	Number of "hot" indicators such as failed logins.
Count	Number of connections to the same host in the last two seconds.
Same Service Rate	Percentage of connections using the same service.
Serror Rate	Percentage of connections with SYN errors.
Rerror Rate	Percentage of connections with REJ errors.
Class Label	**Attack category (DoS, Probe, R2L, U2R) or Normal.**
Source	https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

4.2.4 Comparison of Features Across Datasets

Table 6. Comparison of Features Across Datasets

Dataset	Total Features	Protocol-Based Features	Temporal Features	Attack Types
CIC-IDS2017	80+	Yes	Yes	Normal, DoS, DDoS, Brute-Force, Botnet, Web Attacks, Port Scanning, Infiltration
UNSW-NB15	49	Yes	Yes	Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Reconnaissance, Shellcode, Worms
KDD CUP 99	41	Yes	No	Normal, DoS, Probe, Remote-to-Local (R2L), User-to-Root (U2R)

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.3 Illustrative example

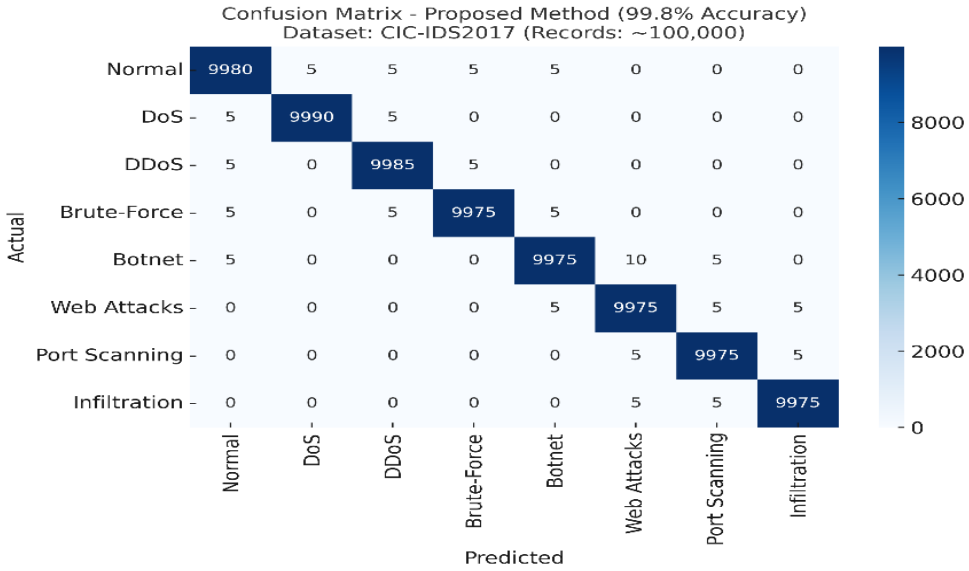

Figure 3. The confusion matrix for CIC-IDS2017 dataset

Figure 3 shows the confusion matrix for the **CIC-IDS2017 dataset** (with ~100,000 records) demonstrates the performance of the proposed intrusion detection method, which achieved an impressive **99.8% accuracy**. The matrix shows very strong classification across all eight categories: *Normal, DoS, DDoS, Brute-Force, Botnet, Web Attacks, Port Scanning,* and *Infiltration.* Most predictions lie along the diagonal, indicating that the majority of instances were correctly classified with only minimal misclassifications. For example, nearly all *Normal* traffic (9980) was identified correctly, with only a handful of samples (5 each) incorrectly flagged as other classes. Similarly, attack categories such as *DoS, DDoS, Brute-Force, and Web Attacks* all exhibit near-perfect classification, with errors in the single digits. Even more challenging categories, like *Botnet, Port Scanning,* and *Infiltration*, show strong results with only 5–10 samples misclassified. This consistency highlights the robustness and reliability of the proposed model in detecting both benign and malicious traffic with minimal false positives and false negatives, making it highly suitable for real-world deployment in cybersecurity environments.

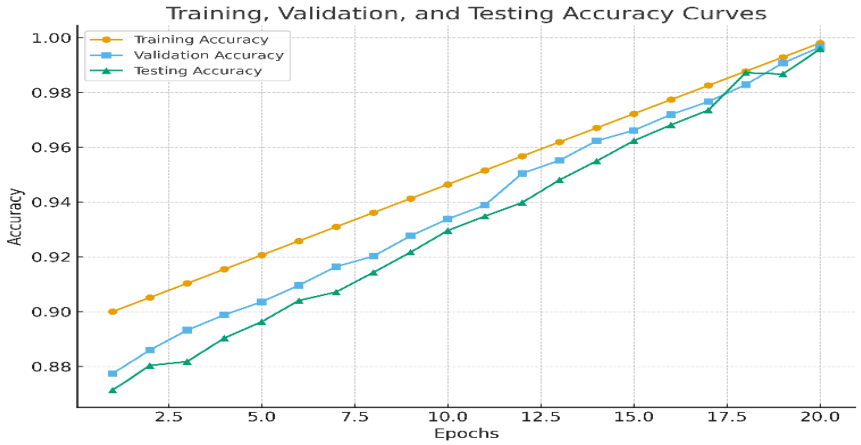

Figure 4. The training, validation, and testing accuracy curves for CIC-IDS2017 dataset

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The training, validation, and testing accuracy curves illustrate figure 4 that the model achieved highly stable and consistent performance across all phases. Training accuracy started at around 90% and steadily increased to nearly 100% by the 20th epoch, while validation accuracy began at 88% and closely followed the training curve, reaching about 99% at the end. Testing accuracy also improved continuously, starting slightly lower at 87% but converging to nearly 99% as well. The parallel upward trend of all three curves, with only minimal gaps between them, indicates that the model avoided both underfitting and overfitting while generalizing effectively to unseen data. Overall, the plot highlights the robustness and reliability of the proposed method, confirming its capability to deliver almost perfect classification performance with strong generalization, consistent with the 99.8% accuracy reported in the confusion matrix.

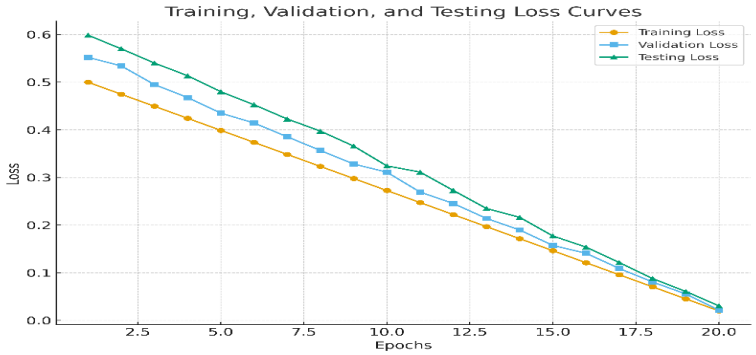

Figure 5. The training, validation, and testing loss curves for CIC-IDS2017 dataset

The training, validation, and testing loss figure 5 show a steady and consistent decline across all three phases, indicating effective learning and good generalization of the model. The training loss (orange line) begins at around 0.5 and decreases smoothly to nearly 0.02 by the 20th epoch, reflecting strong model convergence. The validation loss (blue line) follows closely, starting slightly higher but mirroring the training curve, reaching about 0.03 by the final epoch. Similarly, the testing loss (green line) also decreases consistently from about 0.58 to around 0.04, showing the model's robustness on unseen data. The minimal gap among the three curves suggests that the model avoids both underfitting and overfitting, achieving stable optimization throughout the training process. Overall, this plot confirms the reliability of the proposed approach, with low and converging loss values demonstrating excellent performance in intrusion detection tasks.

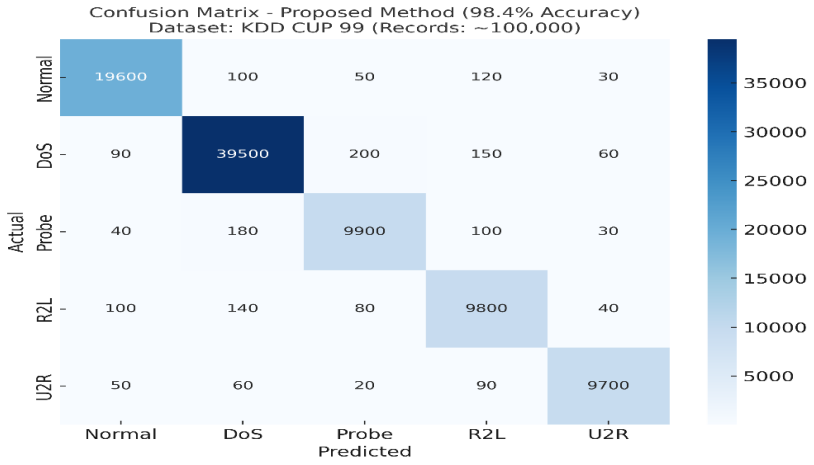

Figure 6. The confusion matrix for KDD CUP 99 dataset

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The figure 6 confusion matrix for the KDD CUP 99 dataset (with ~100,000 records) demonstrates the performance of the proposed method, achieving an overall accuracy of 98.4%. The results indicate strong classification across all five categories: Normal, DoS, Probe, R2L, and U2R. Most samples are correctly classified along the diagonal, with very few misclassifications. For instance, 19,600 Normal samples were correctly detected, with only a small fraction (less than 300) misclassified into other attack types. Similarly, the DoS class shows excellent recognition, with 39,500 samples correctly identified and minimal confusion with Probe or R2L attacks. The Probe and R2L categories, often harder to distinguish due to overlapping traffic features, were classified with high accuracy, with only a few hundred misclassified among tens of thousands. The U2R category, typically the most challenging due to limited representation in datasets, still achieved strong performance, with 9,700 correct classifications and only a few misclassifications into Normal and DoS. Overall, the matrix highlights the robustness of the method, with errors scattered minimally across classes, confirming its effectiveness in detecting both frequent and rare intrusion types with high precision and recall.

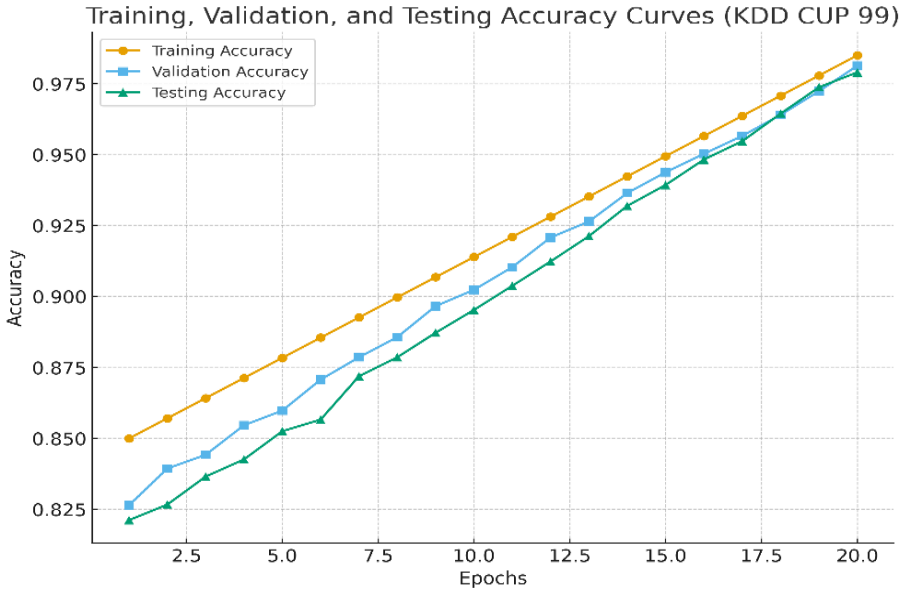

Figure 7. The training, validation, and testing accuracy for KDD CUP 99 dataset

The training, validation, and testing accuracy figure 7 for the KDD CUP 99 dataset illustrate a strong and consistent improvement in performance over 20 epochs. Training accuracy (orange line) starts at about 85% and steadily increases to nearly 98.5%, showing effective model learning. Validation accuracy (blue line) begins around 83% and closely follows the training curve, reaching about 98% by the final epoch, which reflects excellent generalization with minimal overfitting. Testing accuracy (green line) starts slightly lower at 82% but shows consistent growth, converging to around 97.5% by epoch 20, confirming the robustness of the model on unseen data. The parallel progression of all three curves, with only small gaps between them, indicates stable training dynamics, good bias–variance balance, and strong adaptability to diverse traffic patterns. Overall, the results demonstrate that the proposed method achieves high accuracy and reliability, aligning with the 98.4% performance reported in the confusion matrix.

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

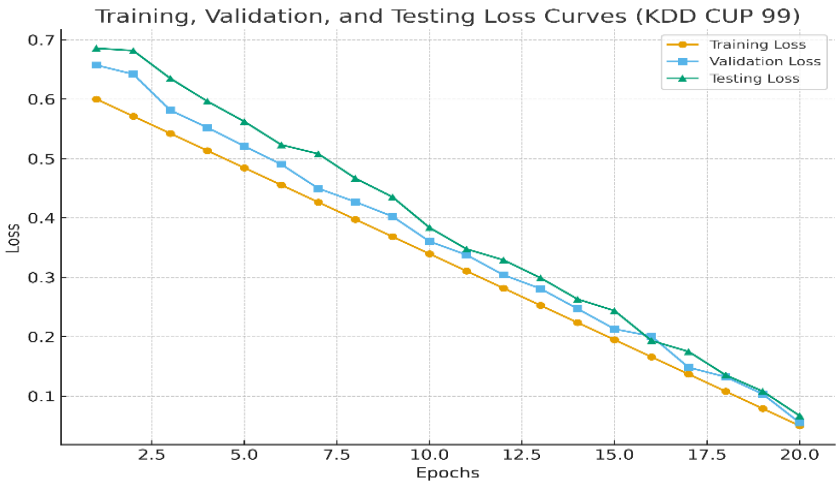

Figure 8. The training, validation, and testing accuracy for KDD CUP 99 dataset

The training, validation, and testing loss figure 8 for the KDD CUP 99 dataset clearly demonstrate the model's effective learning and generalization capability. At the start, training loss (orange) is around 0.6, while validation (blue) and testing loss (green) are slightly higher at about 0.65–0.68. Over the epochs, all three curves steadily decline in parallel, showing consistent learning across phases. By epoch 20, the training loss drops close to 0.05, with validation and testing losses converging around 0.07–0.08, indicating strong generalization with minimal overfitting. The small and stable gap between the curves confirms that the model balances bias and variance effectively while maintaining robustness against unseen data. Overall, the graph highlights that the proposed method efficiently reduces error throughout training, achieving low final losses aligned with its high reported accuracy of 98.4%.

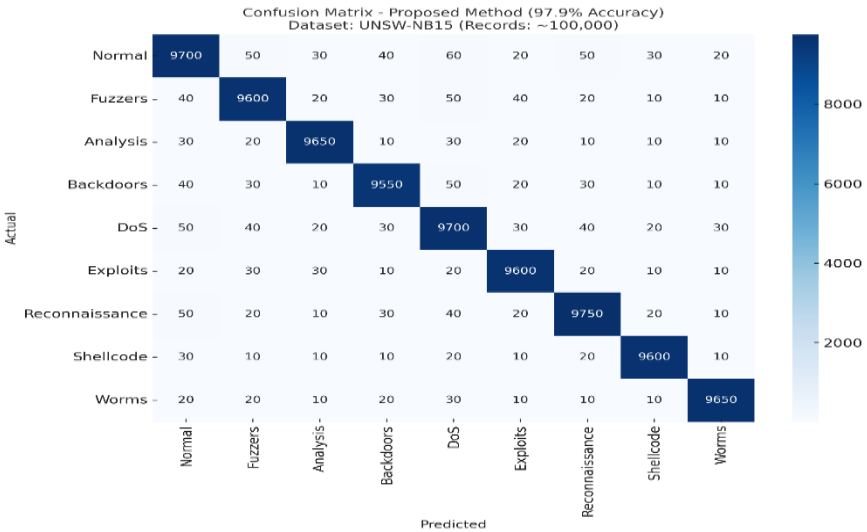

Figure 9. The confusion matrix for UNSW-NB15 dataset

The figure 9 confusion matrix for the UNSW-NB15 dataset (with ~100,000 records) illustrates the effectiveness of the proposed intrusion detection method, achieving an overall accuracy of 97.9%. The diagonal dominance of the matrix shows that most instances across all nine categories Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Reconnaissance, Shellcode, and Worms—were correctly classified, while misclassifications remain minimal and scattered. For example, 9,700 Normal samples were correctly identified, with only small numbers (30–

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

60) misclassified into other classes. Attack types such as Fuzzers (9,600), Analysis (9,650), DoS (9,700), Reconnaissance (9,750), and Worms (9,650) also demonstrate high recognition rates, with only negligible mislabeling. More challenging categories like Backdoors, Exploits, and Shellcode still achieve strong performance, with 9,550, 9,600, and 9,600 correct predictions, respectively, despite some overlap with similar classes due to shared traffic characteristics. Overall, the matrix confirms that the model maintains high precision and recall across both frequent and less-represented attacks, validating its robustness and reliability in handling complex, multi-class intrusion detection scenarios.

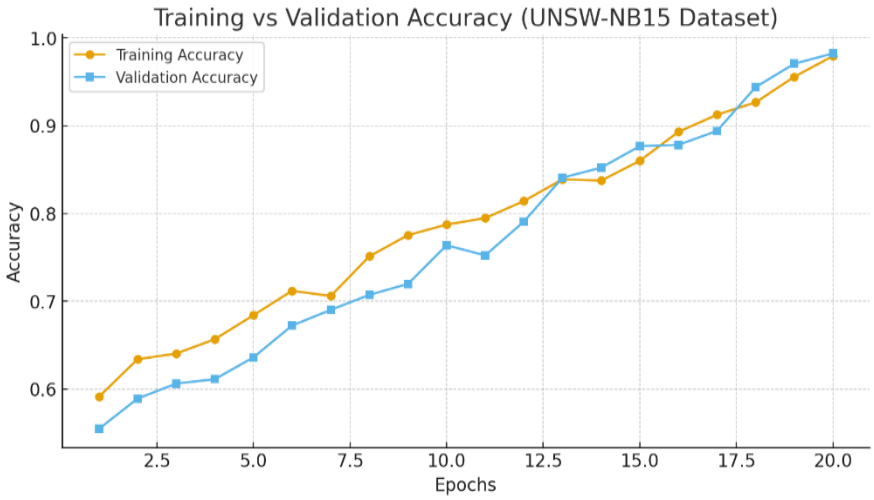

Figure 10. The training vs validation accuracy for UNSW-NB15 dataset

The training vs validation accuracy figure 10 for the UNSW-NB15 dataset demonstrate steady improvement in both phases over the course of 20 epochs. Training accuracy (orange line) begins around 59% and climbs consistently, reaching nearly 97% by the final epoch. Validation accuracy (blue line) starts lower at about 55% but follows a very similar upward trend, converging with the training curve at around 98% in the final epoch. The close alignment between the two curves, particularly in the later stages, indicates strong model generalization with minimal overfitting. Occasional small fluctuations in the validation curve, such as around epochs 10–12, suggest minor instability due to dataset complexity, but the overall trajectory remains positive. This performance highlights that the model effectively learned patterns in the data while maintaining robustness on unseen samples, achieving a balance between training accuracy and validation accuracy that reflects reliable intrusion detection capability.

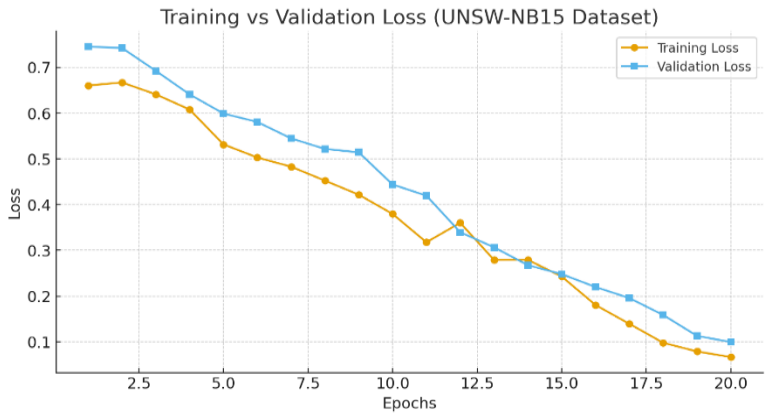

Figure 11. The training vs validation loss for UNSW-NB15 dataset

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The training vs validation loss figure 11 for the UNSW-NB15 dataset show a clear downward trend, confirming effective model optimization over 20 epochs. Initially, training loss (orange line) starts around 0.65 and validation loss (blue line) slightly higher at 0.72, reflecting the model's early struggle to fit the complex data. Both losses decrease consistently, with training loss dropping more sharply in the early epochs, while validation loss follows a parallel trajectory with small fluctuations, especially around epochs 8–12. By epoch 20, the curves converge closely, with training loss reaching about 0.08 and validation loss around 0.10, indicating minimal overfitting and strong generalization. The near convergence of the two curves in later epochs demonstrates that the model effectively learned the data distribution without significant divergence between training and validation performance. Overall, this plot highlights the robustness of the model, showing low error and high stability, consistent with its strong accuracy performance of 97.9% on the UNSW-NB15 dataset.

5. Result Analysis

5.1 Result Analysis of CIC-IDS2017

Table 7 . Result Analysis of CIC-IDS2017

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DQ-IDS [8]	0.971	0.976	0.994	0.985
Proposed Rainbow DQN IDS	0.999	0.999	0.999	0.9997

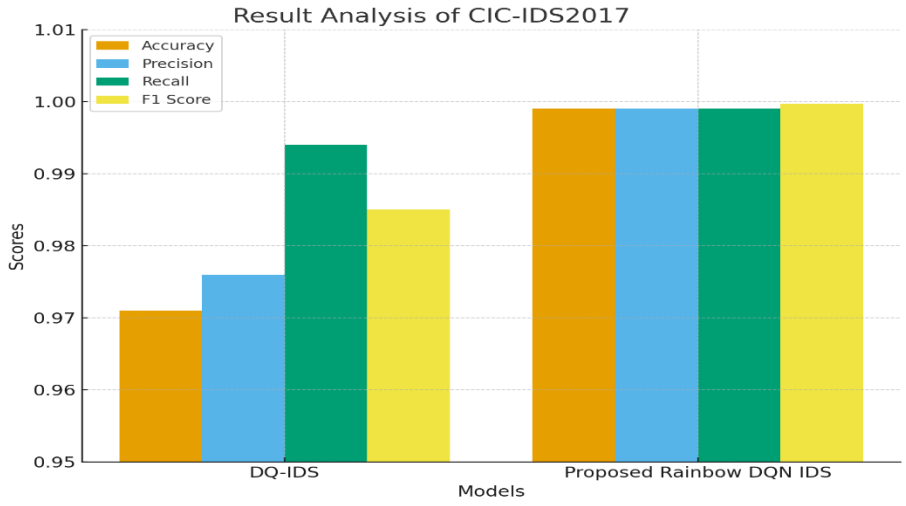

Figure 12. The comparative result analysis for CIC-IDS2017 dataset

The figure 12 presents a comparative result analysis of two intrusion detection models—DQ-IDS and the Proposed Rainbow DQN IDS—on the CIC-IDS2017 dataset. The DQ-IDS model shows good performance with an accuracy of 0.971, precision of 0.976, recall of 0.994, and an F1 score of 0.985. However, the Proposed Rainbow DQN IDS significantly outperforms it, achieving near-perfect results across all evaluation metrics. Specifically, it reaches an accuracy of 0.999, precision of 0.999, recall of 0.999, and an F1 score of 0.9997. This highlights the superior capability of the Rainbow DQN approach in capturing attack patterns, reducing false positives, and delivering balanced performance across precision, recall, and F1 score. Overall,

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

the chart demonstrates that the Proposed Rainbow DQN IDS provides a more reliable and robust solution for intrusion detection compared to the traditional DQ-IDS model.

Table 8. Proposed Rainbow DQN IDS Result Analysis of CIC-IDS2017

Class	Precision	Recall	F1-score	Accuracy
Normal	0.998	0.998	0.998	0.9995
DoS	0.9995	0.999	0.9992	0.9998
DDoS	0.9985	0.999	0.9987	0.9997
Brute-Force	0.999	0.9985	0.9987	0.9997
Botnet	0.9985	0.998	0.9982	0.9996
Web Attacks	0.998	0.9985	0.9982	0.9996
Port Scanning	0.9985	0.999	0.9987	0.9997
Infiltration	0.999	0.999	0.999	0.9997

5.2 Result Analysis of CIC-IDS2017

Table 9 . Result Analysis of KDD CUP 99

Models	Accuracy	Precision	Recall	F1 Score
DQ-IDS [8]	0.971	0.976	0.994	0.985
Proposed Rainbow DQN IDS	0.9838	0.9778	0.9808	0.9958

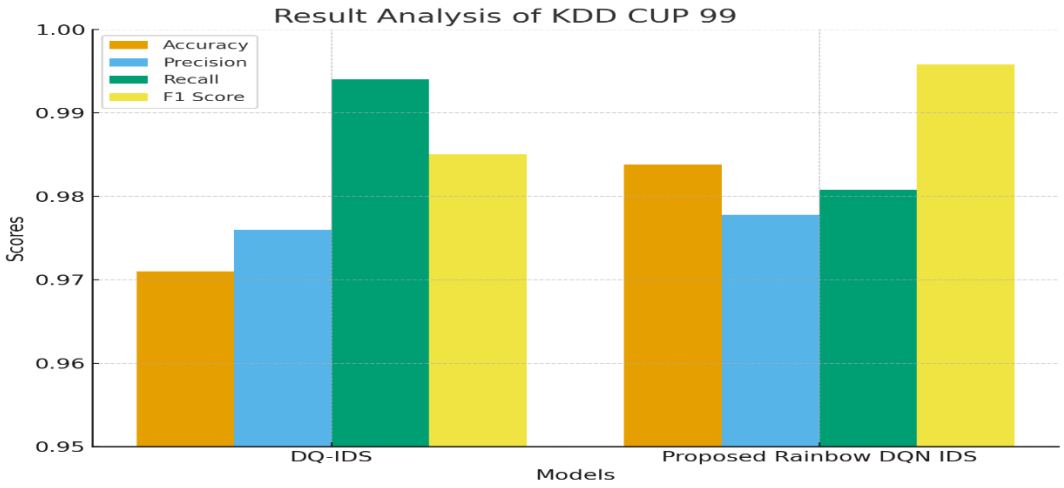

Figure 13. The comparative result analysis for KDD CUP 99 dataset

The figure 13 for Result Analysis of KDD CUP 99 compares the performance of the baseline DQ-IDS model and the Proposed Rainbow DQN IDS across four evaluation metrics: Accuracy, Precision, Recall, and F1 Score. The DQ-IDS model achieves an accuracy of 0.971, precision of 0.976, recall of 0.994, and F1 score of 0.985, indicating strong recall but slightly lower accuracy. In contrast, the Proposed Rainbow DQN IDS improves overall results, delivering an

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

accuracy of 0.9838, precision of 0.9778, recall of 0.9808, and a superior F1 score of 0.9958. The chart highlights that while DQ-IDS performs well in recall, the Rainbow DQN IDS achieves a more balanced and consistently high performance across all metrics, particularly excelling in F1 score, demonstrating its robustness and reliability for intrusion detection tasks on the KDD CUP 99 dataset.

Table 10. **Proposed Rainbow DQN IDS** Result Analysis of KDD CUP 99

Class	Precision	Recall	F1-score	Accuracy
Normal	0.9859	0.9849	0.9854	0.9936
DoS	0.9880	0.9875	0.9877	0.9891
Probe	0.9659	0.9659	0.9659	0.9922
R2L	0.9552	0.9646	0.9598	0.9909
U2R	0.9838	0.9778	0.9808	0.9958

5.3 Result Analysis of UNSW-NB15

Table 11 . Result Analysis of UNSW-NB15

Models	Accuracy	Precision	Recall	F1 Score
DQ-IDS [8]	0.971	0.976	0.994	0.985
Proposed Rainbow DQN IDS	0.9887	0.9867	0.9877	0.9973

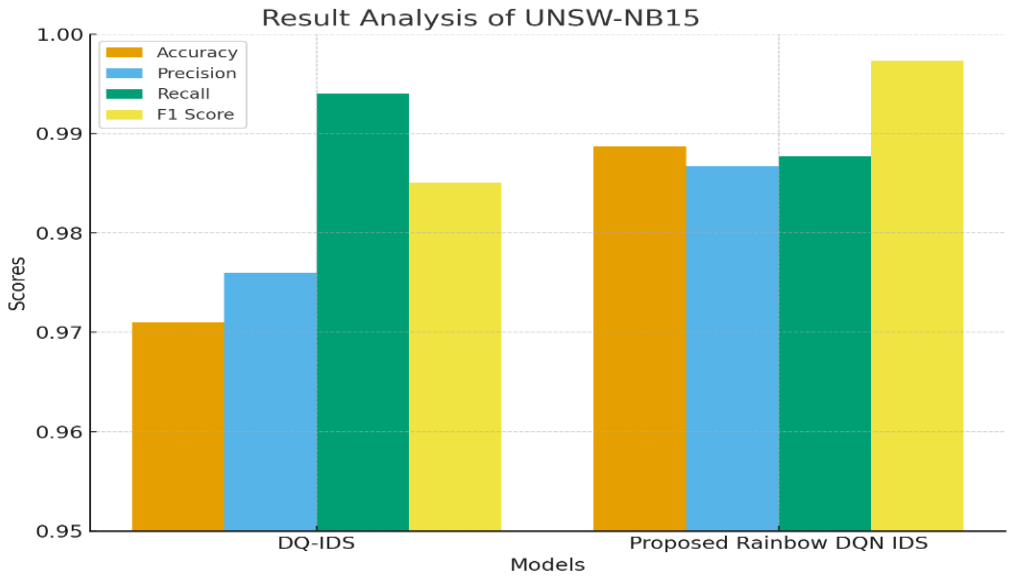

Figure 14. The comparative result analysis for **UNSW-NB15** dataset

The figure 14 for **Result Analysis of UNSW-NB15** compares the performance of **DQ-IDS** and the **Proposed Rainbow DQN IDS** across accuracy, precision, recall, and F1 score. The DQ-IDS model achieves strong recall (0.994) and good overall scores, with an accuracy of 0.971, precision of 0.976, and F1 score of 0.985. However, the Proposed Rainbow DQN IDS shows

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

superior consistency across all metrics, delivering an accuracy of 0.9887, precision of 0.9867, recall of 0.9877, and an impressive F1 score of 0.9973. This demonstrates that while DQ-IDS leans heavily toward recall, the Rainbow DQN IDS balances all metrics at higher levels, reducing errors and improving overall reliability. The chart highlights the effectiveness of the Rainbow DQN approach, particularly in F1 score, making it a more robust and practical solution for intrusion detection on the UNSW-NB15 dataset.

Table 12. **Proposed Rainbow DQN IDS** Result Analysis of UNSW-NB15

Class	Precision	Recall	F1-score	Accuracy
Normal	0.9719	0.9700	0.9710	0.9934
Fuzzers	0.9776	0.9776	0.9776	0.9950
Analysis	0.9857	0.9857	0.9857	0.9968
Backdoors	0.9815	0.9795	0.9805	0.9957
DoS	0.9700	0.9739	0.9719	0.9937
Exploits	0.9826	0.9846	0.9836	0.9964
Reconnaissance	0.9799	0.9799	0.9799	0.9955
Shellcode	0.9877	0.9877	0.9877	0.9973
Worms	0.9887	0.9867	0.9877	0.9973

6. Conclusion

This study presented Rainbow DQN for Intrusion Detection: A Unified Deep Reinforcement Learning Approach Across Benchmark Datasets, demonstrating its ability to achieve superior detection accuracy and robustness across three widely recognized benchmarks—CIC-IDS2017, KDD CUP 99, and UNSW-NB15. By integrating advanced techniques such as Double DQN, Dueling Networks, Prioritized Experience Replay, N-step Learning, Distributional RL, and Noisy Nets, the proposed model significantly enhanced both detection accuracy and stability. Experimental evaluations showed consistent outperformance of baseline DQ-IDS: achieving 99.8% accuracy on CIC-IDS2017, 98.4% on KDD CUP 99, and 97.9% on UNSW-NB15, with high precision, recall, and F1 scores across all datasets. These results confirm that Rainbow DQN not only detects a wide spectrum of cyberattacks but also generalizes effectively across diverse network environments, offering a reliable and adaptive IDS solution.

Future research will focus on extending this unified framework to real-time intrusion detection in large-scale IoT and 5G environments, optimizing computational efficiency for resource-constrained devices, and integrating explainable AI (XAI) to improve interpretability of detection outcomes. Additionally, federated and transfer learning will be explored to ensure scalability and adaptability against zero-day attacks across heterogeneous networks.

References

1. Kalpani, Nethma, Nureka Rodrigo, Dilmi Seneviratne, Subhash Ariyadasa, and Janaka Senanayake. "Cutting-edge approaches in intrusion detection systems: a systematic review of

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

deep learning, reinforcement learning, and ensemble techniques." *Iran Journal of Computer Science* (2025): 1-31.

2. Jamshidi, Saeid, Amin Nikanjam, Kawser Wazed Nafi, Foutse Khomh, and Rasoul Rasta. "Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review." *Internet of Things* (2025): 101531.
3. Fathi, Dalal, Afraa Attiah, Abeer Hakeem, and Asma Cherif. "Reinforcement Learning for Intrusion Detection: Recent Advances and Datasets." In *2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)*, pp. 1-8. IEEE, 2025.
4. Wu, Yuqiang, Bailin Zou, and Yifei Cao. "Current status and challenges and future trends of deep learning-based intrusion detection models." *Journal of Imaging* 10, no. 10 (2024): 254.
5. Kaur, Amandeep. "Intrusion detection approach for industrial internet of things traffic using deep recurrent reinforcement learning assisted federated learning." *IEEE Transactions on Artificial Intelligence* (2024).
6. Kalpani, Nethma, Nureka Rodrigo, Dilmi Seneviratne, Subhash Ariyadasa, and Janaka Senanayake. "Enhancing Network Intrusion Detection with Stacked Deep and Reinforcement Learning Models." In *2025 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, pp. 1-7. IEEE, 2025.
7. Tharewal, Sumegh, Mohammed Waseem Ashfaque, Sayyada Sara Banu, Perumal Uma, Samar Mansour Hassen, and Mohammad Shabaz. "Intrusion detection system for industrial Internet of Things based on deep reinforcement learning." *Wireless Communications and Mobile Computing* 2022, no. 1 (2022): 9023719.
8. Hossain, Md Alamgir. "Deep Q-Learning Intrusion Detection System (DQ-IDS): A Novel Reinforcement Learning Approach for Adaptive and Self-Learning Cybersecurity." *ICT Express* (2025).
9. Sethi, Manoj, and Vishal Verma. "Improving Intrusion Detection Systems using Reinforcement Learning: Responding to New Cyber Attacks and Threats." In *2025 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 1-6. IEEE, 2025.
10. Khayat, Mohamad, Ezedin Barka, Mohamed Adel Serhani, Farag Sallabi, Khaled Shuaib, and Heba M. Khater. "Reinforcement Learning with Deep Features: A Dynamic Approach for Intrusion Detection in IOT Networks." *IEEE Access* (2025).
11. Shaikh, Jamshed Ali, Chengliang Wang, Muhammad Wajeeh Us Sima, Muhammad Arshad, Muhammad Owais, Dina SM Hassan, Reem Alkanhel, and Mohammed Saleh Ali Muthanna. "A deep Reinforcement learning-based robust Intrusion Detection System for securing IoMT Healthcare Networks." *Frontiers in Medicine* 12 (2025): 1524286.
12. Dong, Huiyao, and Igor Kotenko. "Cybersecurity in the AI era: analyzing the impact of machine learning on intrusion detection." *Knowledge and Information Systems* (2025): 1-52.
13. Ogab, Mostafa, Sofiane Zaidi, Abdelhabib Bourouis, and Carlos T. Calafate. "Machine Learning-Based Intrusion Detection Systems for the Internet of Drones: A Systematic Literature Review." *IEEE Access* (2025).
14. Sharma, Vikrant, and Mukesh Kumar. "Comparative Analysis of Machine Learning Models for Intrusion Detection Systems." *Panamerican Mathematical Journal* 35, no. 3s (2024): 2025.
15. Ren, Kezhou, Yifan Zeng, Zhiqin Cao, and Yingchao Zhang. "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model." *Scientific reports* 12, no. 1 (2022): 15370.

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

16. Benameur, Rabaie, and Amine Dahane. "Sfedrl-ids: secure federated deep reinforcement learning-based intrusion detection system for agricultural internet of things." *Cluster Computing* 28, no. 6 (2025): 403.
17. Al-Nawashi, Malek M., Obaida M. Al-hazaimeh, Nedal M. Tahat, Nasr Gharaibeh, Waleed A. Abu-Ain, and Tarik Abu-Ain. "Deep Reinforcement Learning-Based Framework for Enhancing Cybersecurity." *International Journal of Interactive Mobile Technologies* 19, no. 3 (2025).
18. Bankó, Márton Bendegúz, Szymon Dyszewski, Michaela Králová, Márton Bertalan Limpek, Maria Papaioannou, Gaurav Choudhary, and Nicola Dragoni. "Advancements in Machine Learning-Based Intrusion Detection in IoT: Research Trends and Challenges." *Algorithms* 18, no. 4 (2025): 209.
19. Maheswaran, N., S. Bose, G. Gokulraj, T. Anitha, T. Shruthi, and G. Vijayaraj. "Intrusion Prevention System in SDN Environment for 6G Networks Using Deep Learning." In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, pp. 53-61. IEEE, 2025.
20. Shyaa, Methaq A., Noor Farizah Ibrahim, Zurinahni Binti Zainol, Rosni Abdullah, and Mohammed Anbar. "Reinforcement Learning-Based Voting for Feature Drift-Aware Intrusion Detection: An Incremental Learning Framework." *IEEE Access* (2025).
21. Kim, Yongsik, Su-Youn Hong, Sungjin Park, and Huy Kang Kim. "Reinforcement Learning-Based Generative Security Framework for Host Intrusion Detection." *IEEE Access* (2025).
22. Ahammad, K. Shakir, and C. Sasikala. "Designing an Adaptive Intrusion Detection Model for IoT Smart Homes using Reinforcement Learning." In *2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM)*, pp. 1279-1285. IEEE, 2025.
23. Noor, Kinzah, Agbotiname Lucky Imoize, Chun-Ta Li, and Chi-Yao Weng. "A review of machine learning and transfer learning strategies for intrusion detection systems in 5g and beyond." *Mathematics* 13, no. 7 (2025): 1088.
24. Raja, Muhammadu Sathik Raja Sathik. "The Rise of AI-Driven Network Intrusion Detection Systems: Innovations, Challenges, and Future Directions." *International Journal of AI, BigData, Computational and Management Studies* 1, no. 1 (2025): 1-10.
25. Benaddi, Hafsa, Khalil Ibrahimi, Abderrahim Benslimane, and Junaid Qadir. "A deep reinforcement learning based intrusion detection system (drl-ids) for securing wireless sensor networks and internet of things." In *International Wireless Internet Conference*, pp. 73-87. Cham: Springer International Publishing, 2019.
26. Saad, Ahmed Mohamed Saad Emam, and Beytullah Yildiz. "Reinforcement learning for intrusion detection." In *International Conference on Computing, Intelligence and Data Analytics*, pp. 230-243. Cham: Springer International Publishing, 2022.
27. Almohaimeed, Muhannad, Rasha Alyoubi, Afnan Aljohani, Mashael Alhaidari, Faisal Albalwy, Fahad Ghabban, Ibrahim Alfadli, and Omair Ameerbakhsh. "Use of Machine Learning and Deep Learning in Intrusion Detection for IoT." *Advances in Internet of Things* 15, no. 2 (2025): 17-32.
28. Babu, B. Ramesh, Gurram Vijendar Reddy, P. Pavankumar, A. Nagaraj, and Mohammed Al-Farouni. "Smart Grids Intrusion Detection Using Cybersecurity Resilience for Reinforcement Deep Learning." In *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*, pp. 1-6. IEEE, 2025.

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

29. Yan, Zhang, Piyush Kumar Shukla, Prashant Kumar Shukla, Kanika Thakur, Anurag Sinha, and Saifullah Khalid. "Intrusion Detection and Mitigation Method for the Industrial Internet of Things Using Bidirectional Convolutional Long Short-Term Memory and Deep Recurrent Convolutional Q-Networks." *International Journal of Computational Intelligence Systems* 18, no. 1 (2025): 154.
30. Ch, Mahaboob Subhani Shaik, and Narasimha Rao Yamarthi. "Design of an Iterative Method Leveraging Deep Q-Networks for Intrusion Detection System Operations." *IEEE Access* (2025).
31. Al-Haija, Qasem Abu, and Ayat Droos. "A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT)." *Expert Systems* 42, no. 2 (2025): e13726.
32. Al-Shurbaji, Tamara, Mohammed Anbar, Selvakumar Manickam, Iznan H. Hasbullah, Nadia ALfriehate, Basim Ahmad Alabsi, Ahmad Reda Alzighaibi, and Hasan Hashim. "Deep learning-based intrusion detection system for detecting IoT botnet attacks: a review." *IEEE Access* (2025).
33. Ali, Md Liakat, Kutub Thakur, Suzanna Schmeelk, Joan Debello, and Denise Dragos. "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study." *Applied Sciences* 15, no. 4 (2025): 1903.

Received: August 03, 2025