Article

Not peer-reviewed version

Real-Time DDoS Detection in Industrial IoT Using Proximal Policy Optimisation and Deep Reinforcement Learning

Mikiyas Alemayehu , Mohamed Chahine Ghanem * , Hamza Kheddar , Dipo Dunsin , Chaker Abdelaziz Kerrache , Geetanjali Rathee

Posted Date: 2 January 2026

doi: 10.20944/preprints202601.0081.v1

Keywords: PPO; IIoT; SCADA; DDoS detection; deep reinforcement learning; real-time detection; IDS; ONNX; Critical National Infrastructures; OT security

 doi:10.20944/preprints202601.0081.v1

Article

Real-Time DDoS Detection in Industrial IoT Using Proximal Policy Optimisation and Deep Reinforcement Learning

Mikiyas Alemayehu [1], Mohamed Chahine Ghanem [1,2,*], Hamza Kheddar [3], Dipo Dunsin [4], Chaker Abdelaziz Kerrache [5] and Geetanjali Rathee [6]

[1] Cyber Security Research Centre, London Metropolitan University, London, N7 8DB, UK
[2] Cybersecuirty Institute, University of Liverpool, Liverpool, L69 3BX, UK
[3] LSEA Laboratory, Department of Electrical Engineering, University of Medea, Medea, Algeria.
[4] Institute of Inner City Learning, University of Wales Trinity Saint David, London, E14 4HA, UK
[5] Laboratoire d'Informatique et de Mathématiques, University Amar Telidji Laghouat, Laghouat, Algeria.
[6] Department of Computer Science and Engineering, Netaji Subhas University of Technology, New Delhi, 110078, India.
* Correspondence: Dr MC. Ghanem (mohamed.chahine.ghanem@liverpool.ac.uk).

Abstract

Industrial Internet of Things (IIoT) and SCADA-connected networks face disruptive DDoS events where detection must be both accurate and low-latency at the edge. This study benchmarks deep reinforcement learning (DRL) for real-time binary attack detection and proposes a Proximal Policy Optimisation (PPO) detector tailored for deployment. Five DRL agents—DQN, Double DQN, Duelling DQN, DDPG, and PPO—are trained under a unified preprocessing pipeline (automatic label mapping, numeric-feature selection, robust scaling, and class balancing) and evaluated on three representative datasets: KDDCup99, CIC-DDoS2019, and Edge-IIoTset. We report accuracy, precision/recall, F1-score, false-positive/false-negative rates, and AUC-ROC, alongside CPU latency to reflect operational constraints. Across all datasets, PPO achieves the best accuracy–latency trade-off, reaching 99.3% accuracy on KDDCup99, 93.7% on CIC-DDoS2019, and 95.5% on Edge-IIoTset, while maintaining inference latency below 0.23 ms per sample. PPO also converges faster and is more sample-efficient than value-based alternatives. For practical adoption, the trained PPO policies are exported to ONNX (one model per dataset), enabling lightweight, PyTorch-independent inference on resource-constrained industrial gateways.

Keywords: PPO; IIoT; SCADA; DDoS detection; deep reinforcement learning; real-time detection; IDS; ONNX; Critical National Infrastructures; OT security

1. Introduction

Industrial Internet of Things (IIoT) infrastructures are rapidly expanding across various sectors, including energy and power systems such as smart grids, wind and solar farms, and oil and gas pipelines and other Critical National Infrastructures (CNIs) that are based on IIoT sensors, Supervisory Control and Data Acquisition (SCADA) systems, and pressure/flow sensors for remote monitoring and real-time telemetry [1,2]. Predictive maintenance of transformers, turbines, and pipelines is also enabled by IIoT. This adoption is not limited to energy; it spans water and waste management, manufacturing and industrial automation, transportation, logistics, healthcare, defence, smart cities, and public infrastructures [3].

This widespread integration dramatically increases interconnectivity and expands the cyber attack surface. DDoS attacks remain one of the most prevalent and disruptive threats, which is capable of disrupting power grids, water treatment plants, and manufacturing systems within seconds [4,5]. Unlike conventional IT/OT environments, IIoT systems are inherently cyber-physical, coupling network communication with real-world sensors and actuators. Consequently, a successful DDoS

 doi:10.20944/preprints202601.0081.v1

attack not only consumes computational resources and bandwidth, but also produces measurable effects on physical parameters [5,6].

Traditional signature-based and threshold-based Intrusion Detection Systems (IDSs), designed primarily for enterprise or cloud contexts, fail to capture these intertwined cyber-physical dynamics and suffer from high false-positive rates (FPR), as well as complete failure against zero-day attacks [7]. Machine Learning (ML)-based approaches, such as Random Forest (RF) and Support Vector Machines (SVM), rely on static packet features, while recurrent models like Long-Short-Term-Memory (LSTM) networks, which are sequence-aware, impose heavy computational demands and high latency, which is unsuitable for delay-sensitive critical systems [8]. Recent advancements in DRL (DRL) show promise for dynamic cybersecurity problems, enabling agents to adaptively learn optimal detection policies in uncertain, evolving environments [9,10]. However, most DRL-based solutions remain experimental, and they lack deployment-oriented implementations with real-time inference, edge optimisation, and interoperability [11,12].

Recent studies have explored DRL for adaptive DDoS mitigation, reporting high accuracies with inference latencies exceeding 1 ms, which renders them unacceptable for real-time protection of critical systems [13,14].

This paper addresses these gaps by presenting a comprehensive multi-model, multi-dataset evaluation of DRL for real-time DDoS detection in IIoT environments. Five state-of-the-art DRL agents: DQN, Double DQN, Duelling DQN, DDPG, and PPO are systematically trained and compared across three benchmark datasets: KDDCup99, CIC-DDoS2019, and Edge-IIoT. A unified preprocessing pipeline with class balancing and robust scaling is applied consistently. Results demonstrate that PPO significantly outperforms the others, achieving 99.3% accuracy on KDDCup99, 93.7% on CIC-DDoS2019, and 95.5% on Edge-IIoT, with inference latency below 0.23 ms per sample and low false positives/negatives.

The main contributions of this work are as follows:

1. A robust, unified preprocessing pipeline applied across three diverse benchmark datasets, enabling fair multi-model comparison.
2. Comprehensive evaluation of five DRL agents, establishing PPO as the superior method for DDoS detection in IIoT environments in terms of accuracy, convergence speed, and real-time performance.
3. End-to-end benchmarking including accuracy, precision, recall, F1-score, FPR/NR, inference latency, and throughput.
4. Export of the best-performing PPO models (one per dataset) to Open Neural Network Exchange (ONNX) format, enabling lightweight, PyTorch-independent deployment on resource-constrained industrial gateways.

The remainder of this paper is structured as follows: Section 2 discusses related work; Section 3 details the methodology and DRL training pipeline; Section 4 presents experimental results and evaluation metrics; Section 5 discusses findings and limitations; and Section 6 concludes with future research directions.

2. Related Work

RL and DRL have been increasingly explored for intrusion and DDoS detection, particularly in IoT/IIoT settings, with most works evaluated using confusion-matrix-based metrics such as accuracy, precision, recall, F1-score, FPR and FNR.

Based on the survey work, Gueriani et al. [15] provide a focused survey on DRL-based intrusion detection for IoT environments. They classify existing systems by application domain, such as wireless sensor networks, DQN-based schemes, healthcare, hybrid, and miscellaneous approaches and by the underlying DRL algorithm. The authors synthesise how these systems are designed: state, action, reward, network architecture and how they are evaluated. The result is a clear taxonomy showing that accuracy, precision, recall, FNR, FPR, and F-measure are the dominant metrics used to compare

 doi:10.20944/preprints202601.0081.v1

DRL-based IDS models. They give a compact yet detailed map of DRL-IDS design choices and performance trends. However, they do not implement or benchmark new models themselves, so no new confusion matrices or datasets are introduced. Instead, they report on many existing datasets, including NSL-KDD, CICIDS, AWID, and IoT botnet traces used in the surveyed works.

Yang et al. [16] review DRL techniques for network intrusion detection more broadly, including but not limited to IoT. They focus on what DRL adds compared to conventional deep learning, particularly in handling sequential decision-making, online adaptation, and partial observability. They highlight technical challenges such as training efficiency, detecting minority and unknown attack classes, feature selection, and class imbalance, and systematically compare reported accuracy, recall, precision, and F1-score between DRL and non-DRL models. The strength of this survey includes its algorithmic depth. It dissects how DQN, Double DQN, duelling networks, and actor-critic families are instantiated in NIDS. However, just like Gueriani et al. ([15]), it is descriptive rather than experimental and uses reported results from other papers.

A complementary survey in SN Computer Science in [17] systematically categorises RL-based IDS approaches across different environments, including enterprise networks, IoT, and cloud. The authors analyse RL algorithms, including Q-learning, SARSA, DQN, and actor-critic, which are used, observations such as packet/flow features, host-level metrics they consume, and actions such as alerting, rate-limiting, and moving-target defence that they take. Their results are a structured taxonomy and a summary of performance ranges, reported in terms of accuracy, detection rate, precision, and F1-score. The strength here is the unifying view. The paper shows where RL has clear benefits, including adaptive thresholding and sequential defence policies. However, it does not run new experiments, and performance comparisons are indirect because different works use different datasets and label granularities. As in previous surveys, common datasets discussed include NSL-KDD, KDD'99, and CICIDS2017.

A more general survey on cyber security and RL [18] covers IDS/IPS, IoT, and identity and access management (IAM) systems. It identifies key use-cases such as anomaly-based NIDS, adaptive firewalls, moving-target defence, access-control policy optimisation and summarises how RL models are trained and evaluated. The reported result is that RL-based IDS typically achieve competitive or improved detection rate and accuracy relative to static baselines on datasets such as NSL-KDD, CICIDS, and AWID. The strength is to broaden the view beyond IDS into other control-plane tasks. However, it lacks a unified benchmark or head-to-head comparison, so conclusions about absolute performance are qualitative. The paper aggregates results from a broad set of existing corpora.

In [19], the authors introduce a transformer-based federated Double Q-Network framework for dynamic intrusion detection in IoT networks. Conceptually, it uses a transformer encoder as a federated aggregator to capture global cross-client patterns, while a dual-layer Q-network separates feature extraction from decision optimisation on each client. Soft Actor-Critic (SAC) is used locally to handle non-IID data and client heterogeneity. The results reported include detection accuracies above 99% on multiple NetFlow-based datasets, including NF-BoT-IoT, NF-ToN-IoT, NF-CSE-CIC-IDS2018-v2, NF-UNSW-NB15-v2 and additional UNSW-MG24 and CIC IIoT 2025 datasets, along with improved F1-scores and reduced FPR compared to centralised or non-transformer baselines. Strong generalisation across heterogeneous datasets and robustness to non-IID client data has been demonstrated. However, the limitations include increased model complexity, higher communication overhead in federated training, and the fact that evaluations are limited to lab-scale environments rather than large production IIoT systems.

Rashid et al. [20] designed a federated learning-based approach for intrusion detection in IIoT networks, motivated by the privacy and governance limitations of centralised machine learning. Their method trains local models on the Edge-IIoTset dataset, which is an IIoT-specific dataset containing DoS/DDoS, information gathering, MITM, injection, and malware attacks, then aggregates parameters on a central server. They report that their federated global model achieves 92.49% accuracy, very close to their centralised baseline, and comparable precision, recall and F1-scores, which indicates that

 doi:10.20944/preprints202601.0081.v1

privacy-preserving training does not dramatically degrade detection quality. Privacy preservation has received serious attention in this work. Raw IIoT traffic never leaves the premises, yet global performance remains high. However, there exists communication overhead between clients and server, sensitivity to client participation patterns, and evaluation on a single dataset, which is Edge-IIoTset; therefore, cross-domain generality is not demonstrated.

The authors in [21] propose SFedRL-IDS, a federated DRL-based IDS in the agricultural IoT context. The authors combine federated learning with DRL agents deployed across distributed agricultural IIoT nodes. Each node locally learns to detect anomalies using DRL while parameter updates are aggregated securely to form a global model. They show that this approach yields higher accuracy than non-federated DRL or centralised baselines on IoT intrusion datasets, especially under poisoning and unreliable-client scenarios. The strengths are resilience to adversarial or malfunctioning clients and preservation of local data privacy. However, experiments are conducted mostly with synthetic and small-scale real agricultural traces, and the architectural and energy overhead on constrained field devices is not deeply analysed.

In the healthcare domain, the authors in [22] target the Internet of Medical Things (IoMT) by combining CNN and LSTM for feature extraction with DRL decision-making via DQN and PPO. The system first performs Enhanced Mutual Information Feature Selection, then feeds the selected features to a CNN-LSTM backbone. Finally, a DRL layer, which uses DQN and PPO, refines decisions. Evaluation is conducted on the CICIoMT2024 dataset, it attains a binary-classification accuracy of 99.58 % and a multi-class accuracy of 77.73 % across 18 benign/attack classes, which demonstrates strong binary but moderate multi-class performance. This approach achieves near-perfect detection in binary settings, and its integration of temporal and spatial features with DRL is successful. However, the performance is low on rare attack classes, and it is tailored to IoMT traffic, with no direct evaluation on industrial IoT or DDoS-heavy datasets.

Shi et al. [23] introduce ID-RDRL, which combines recursive feature elimination and a decision-tree classifier with DRL to build an intrusion-detection model. They first apply recursive feature elimination (RFE) with a tree-based model to remove about 80% of redundant features from the CSE-CIC-IDS2018 dataset. Then, they use a DRL agent with policy, value, and Q-function networks to learn policies that maximise detection rewards. Their reported results include an accuracy of around 96.2% with balanced precision and recall over multiple attack classes, including DoS/DDoS. The efficient feature reduction, which leads to faster inference and strong detection performance across known and some previously unseen attacks make the approach promising. However, sensitivity to hyperparameters and evaluation limited to general-purpose CSE-CIC-IDS2018 rather than IIoT-specific corpora, makes the accuracy questionable.

In [24], the authors address learning moving-target defence (MTD) strategies by combining federated and reinforcement learning (FRL) in IoT platforms. Rather than directly classifying packets, their approach learns which configuration changes, i.e. MTD actions to perform in response to anomalies detected by a separate analytics module. Its FRL agents choose MTD techniques based on device fingerprinting and anomaly-detection feedback. The results show faster learning and improved attack-mitigation rates compared to centralised RL, as well as robustness when some clients behave maliciously. The main strength is that it operates on real hardware. Ten heterogeneous IoT devices are used in the testbed. However, this work is not focused solely on DDoS; coverage spans various malware and intrusion behaviours, and its performance is measured more in mitigation success and learning time than in accuracy on a specific labelled dataset, which is a testbed in a custom IoT environment with real devices.

López-Martín et al. [25] reformulate supervised IDS training as a DRL task, replacing the live environment with a "pseudo-environment" that samples labelled records and generates rewards from detection errors. They test four DRL variants: DQN, Double DQN (DDQN), policy gradient, and actor-critic on NSL-KDD and AWID datasets. The results show that DDQN yields the best detection performance, with accuracy and F1-scores that match or exceed classic ML baselines, including SVM,

 doi:10.20944/preprints202601.0081.v1

Random Forest, while offering faster inference than deeper architectures. A clean DRL formulation of a supervised problem and the systematic comparison across four RL algorithms are demonstrated in this work. The age and characteristics of the datasets used in this work do not fit NSL-KDD, and AWID are not IIoT-specific, and the approach lacks evaluation on modern, highly imbalanced IoT traffic.

Another DRL-based NIDS, "Network Intrusion Detection using Deep Reinforcement Learning," [26] proposes a relatively simple policy network and evaluates several RL variants on multiple intrusion datasets. The authors demonstrate that DRL can improve detection rate and accuracy over non-RL baselines and support online adaptation. However, the architecture and experimental details are lighter than in López-Martín et al. Strengths include architectural simplicity, which makes it easier to deploy and achieve gains in detection metrics on benchmark datasets. The work does not deeply analyse low-rate DDoS, IIoT resource constraints, or latency and energy overhead.

Complementing these general and federated approaches, prior contributions focus more directly on DRL-based DDoS detection and mitigation in SDN and IoT/IIoT environments. A flexible SDN-based framework for slow-rate DDoS attack mitigation combines a deep-learning-based IDS with a DRL-based IPS to counter Slow HTTP Read and similar low-rate attacks [27]. The IDS that is typically CNN/LSTM-based detects anomalies, and a DRL agent then learns mitigation policies such as flow-rule installation or rate-limiting in an SDN controller. Emulation results in an SDN data-centre testbed show very high detection accuracy, on the order of the high-90% range, and near-perfect DDoS mitigation rate, with limited impact on benign traffic. The evaluation is expressed using confusion-matrix metrics and mitigation ratios. The end-to-end coverage, including detection and mitigation, explicit targeting of stealthy slow-rate attacks and compatibility with SDN are promising. However, the evaluation is only in an emulated data-centre setting and no explicit consideration of highly resource-constrained IoT gateways. The dataset and traffic are based on CICDDoS-style traces plus synthetic workloads in Mininet/ONOS.

A collaborative stealthy DDoS detection method at the IoT edge employs Soft Actor-Critic to dynamically tune lightweight unsupervised detectors deployed on gateways [13]. In this design, each gateway runs a fast histogram-based outlier score (HBOS) classifier, while a central RL agent learns optimal parameter settings, such as the number of bins and the threshold, based on detection quality feedback from multiple gateways. The results include high detection accuracy for both volumetric and low-rate IoT-based DDoS in experiments using N-BaIoT and a real IoT testbed, while keeping CPU utilisation on gateways under about 2%. The approach is lightweight on-device workload and provides early detection close to the attack source. However, it relies on a central edge server for coordination.

In [28], an IEEE P2668-compliant multi-layer IoT-DDoS defence system (DRL-MLDS) further integrates DRL with standardised IoT performance indicators. The system learns to classify and mitigate multi-protocol DDoS attacks, including ICMP, TCP SYN, UDP, HTTP, MQTT, and CoAP floods, using DRL to drive per-IP blocking decisions. The results show single-protocol detection accuracies above 96% and multi-layer DDoS detection around 97%, with low FPR and improved applicability index (ADex) values that meet or exceed IEEE P2668's recommended thresholds. The multi-layer protocol coverage and integration with an IoT readiness standard are promising. Limitations are specialised testbed assumptions and training overhead that may be non-trivial for very constrained devices. The testbed used in this work is a custom multi-layer IoT DDoS environment published as an IEEE DataPort dataset.

Overall, from generic DRL-based NIDS to specialised IoT/IIoT, federated, and DDoS-focused frameworks, the pattern is consistent. RL and DRL models are trained and compared primarily through confusion-matrix metrics such as accuracy, precision, recall, F1-score, FPR, and FNR. In many cases, ID-RDRL, HCLR-IDS, FedT-DQN, and DDoS-oriented systems reported accuracies are above 95%, with notable gains over non-RL baselines in detecting or mitigating DDoS and related attacks [19,22,23,27].

doi:10.20944/preprints202601.0081.v1

Table 1. RL/DRL-based approaches for intrusion and DDoS detection in IoT/IIoT environments

Ref.	Year	Approach	Product/System	Achievement	Limitations	Dataset/Testbed
[15]	2023	DRL-based IDS survey	IoT intrusion detection landscape	Comprehensive taxonomy of DRL-based IDS	No implementation or report of its own confusion matrix	NSL-KDD, CICIDS, andAWID
[16]	2024	DRL NIDS survey	DRL network intrusion detection	Classifies DRL NIDS by algorithm, state/action/reward design.	No unified experimental comparison or no new model	–
[17]	2024	RL-based IDS survey	RL approaches for IDS	Systematic review of Q-learning, DQN, actor–critic etc.	Lacks head-to-head benchmarks; largely conceptual	NSL-KDD, and CIC family
[18]	2022	RL for cyber-security survey	IDS/IPS, IoT, IAM	Brief overview of RL in security. discusses the use of various metrics.	Limited depth on IoT/IIoT; no concrete evaluation.	Multiple cybersecurity datasets surveyed.
[19]	2025	Transformer-based federated Double Q-network (FedT–DQN)	Collaborative IoT IDS	>99% detection accuracy over centralised baselines	High training complexity, communication overhead in FL.	NF-BoT-IoT, NF-ToN-IoT, NF-CSE-CIC-IDS2018-v2, NF-UNSW-NB15-v2, UNSW-MG24, CIC IIoT 2025
[20]	2023	Federated learning with conventional ML	IIoT IDS for Edge–IIoTset	Global FL model reaches 92.49 % accuracy, close to centralised training, for multi-class.	Accuracy slightly below centralised ML. assumes reliable server	Edge-IIoTset
[21]	2024	Secure federated DRL (SFedRL–IDS)	Agricultural IoT IDS	Higher detection accuracy and better robustness than non-federated DRL.	Evaluated mainly in simulated settings.	Generic sensor-network traces
[22]	2025	CNN–LSTM & DRL (DQN/PPO)	HCLR-IDS for IoMT healthcare	Binary classification accuracy 99.58%. 18-class multi-class accuracy 77.73%.	Multi-class performance is notably lower for rare attack types.	CICIoMT2024 intrusion dataset
[23]	2022	Recursive feature elimination & DRL (policy/value/Q-networks)	ID-RDRL IDS	Accuracy 96.2% on CSE-CIC-IDS2018. removes ≈ 80% of features with minimal performance loss	Trained on a single dataset.	CSE-CIC-IDS2018
[24]	2023	Federated reinforcement learning for moving-target defence	CyberForce framework for malware mitigation	Faster learning and higher mitigation success than centralised RL	Not DDoS-specific, focuses on generic malware.	Testbed of 10 heterogeneous IoT devices infected by malware
[25]	2020	DQN, DDQN, policy gradient, actor–critic	Supervised DRL-based NIDS	DDQN, achieves competitive or better accuracy than classic ML, reduced inference time vs. deep networks	Uses offline labelled data, sensitive to dataset bias	NSL-KDD and AWID

doi:10.20944/preprints202601.0081.v1

Table 1. *Cont.*

Ref.	Year	Approach	Product/System	Achievement	Limitations	Dataset/Testbed
[26]	2023	Deep RL for NIDS	Generic network intrusion detector	Improves detection rate and accuracy over non-RL baselines on benchmark NIDS datasets	Limited details on low-rate DDoS and IIoT scenarios	NSL-KDD and CICIDS
[27]	2023	DL-based IDS & DRL-based IPS for slow-rate DDoS	SDN data-center slow HTTP DDoS mitigation	High detection accuracy and near-100% mitigation success in emulated SDN DC. Minimal impact on benign flows	Tuned for HTTP slow-rate attacks	CICDDoS2019 & Mininet/ONOS data-center testbed
[13]	2023	SAC-tuned HBOS & collaborative RL	Collaborative IoT edge DDoS detector	Achieves ≈ 99% accuracy on volumetric and stealthy low-rate IoT-based DDoS with CPU utilisation.	Requires a central RL server at the edge. HBOS still relies on hand-crafted features	N-BaIoT dataset & real IoT testbed
[28]	2023	DRL-MLDS (IEEE P2668-compliant multi-layer defence)	Multi-layer IoT DDoS defence system	Accuracy > 96% for single-protocol such as ICMP, TCP SYN, UDP, HTTP, MQTT, CoAP floods and ≈ 97% for multi-layer attacks.	Tested on a controlled IEEE DataPort testbed. mainly IoT protocols	Custom multi-layer IoT-DDoS testbed & released dataset (IEEE DataPort)

 doi:10.20944/preprints202601.0081.v1

3. Methodology

The detection pipeline implemented in this work follows a sequential, modular design comprising data ingestion, unified feature engineering, normalisation, reinforcement learning setup, multi-model agent training, evaluation, and model export. The design ensures stability, interoperability, and deployability across diverse IIoT environments while maintaining real-time performance constraints as illustrated in figure 1.

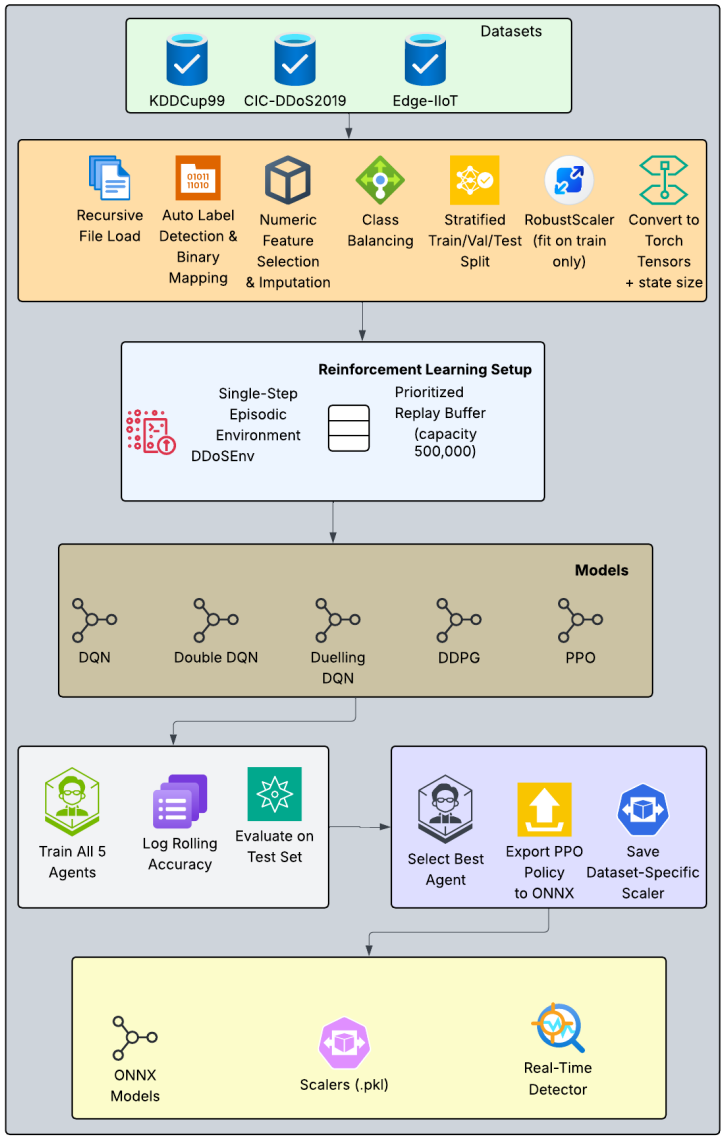

Figure 1. Overview of the multi-model, multi-dataset DRL-based IIoT DDoS detection pipeline

3.1. Datasets

To ensure robust generalisation and enable a fair multi-model comparison, three widely recognised benchmark datasets are employed: KDDCup99, CIC-DDoS2019, and Edge-IIoT. These datasets collectively span diverse network behaviours, attack types, and scales, which make them highly relevant to real-world IIoT security scenarios.

- **KDD Cup 1999**
  The KDD Cup 1999 dataset remains a foundational benchmark. It comprises approximately 4.9 million simulated connections from the 1998 DARPA evaluation, featuring 41 attributes from TCP/IP headers and payload statistics. Attacks are grouped into DoS, Probe, R2L, and U2R categories, with severe imbalance (≈80 % normal). Numeric features are retained (39 after

 doi:10.20944/preprints202601.0081.v1

preprocessing), and undersampling is applied to achieve ≈30 % attack ratio, which yields 138,967 balanced samples suitable for edge evaluation [29].

- **CIC-DDoS2019**
  The CIC-DDoS2019 dataset provides a modern representation of reflection/amplification attacks captured in 2019. It includes 12 contemporary attack types alongside benign traffic, with 78 engineered flow-level features. After incremental loading and undersampling to ≈30 % attack ratio, 139,758 balanced samples are obtained [30].
- **Edge-IIoTset**
  The Edge-IIoTset dataset is a comprehensive collection for IIoT security, which features 15 attack families on a multi-layer testbed. It provides 44 features from traffic and logs, with a natural attack ratio of ≈31 %. The full balanced portion is retained (test split: 356,551 samples) [31].

3.2. Data Preprocessing

Figure 2 is applied consistently across all three datasets to ensure fair comparison and reproducibility. The pipeline comprises the following steps:

Recursive File Loading and Concatenation: All CSV and Parquet files within the dataset-specific directories are discovered recursively and loaded incrementally to manage memory efficiently, particularly for large collections such as CIC-DDoS2019, which contains multiple files and Edge-IIoTset, which contains two large files. The loaded DataFrames are concatenated into a single unified DataFrame per dataset.

Automatic Binary Label Mapping: The label column is automatically detected using common names (e.g., 'Label', 'Attack_type'...). Labels are mapped to binary classes (normal/benign = 0, attack/DDoS = 1) via keyword-based rules, to accommodate variations across datasets.

Numeric Feature Selection with Imputation: Only numeric features are retained after dropping the original label column. Missing and infinite values are replaced with zeros to ensure compatibility with downstream tensor operations.

Class Balancing by Under-sampling: KDDCup99 and CIC-DDoS2019 are highly imbalanced. Under-sampling of the majority, which is the normal class, is applied to achieve approximately 30 % attack ratio, in order to prevent artificially inflated accuracy from skewed distributions. Edge-IIoTset exhibits a more balanced natural distribution and is used without further down-sampling.

Stratified Train/Validation/Test Split: A stratified 70%/15%/15% split is performed with a fixed random seed, preserving class proportions in each subset for reliable evaluation.

Normalisation with `RobustScaler`: The `RobustScaler` is fitted exclusively on the training split and applied to validation and test splits. This scaler, based on the interquartile range, provides robust handling of outliers prevalent in network traffic data.

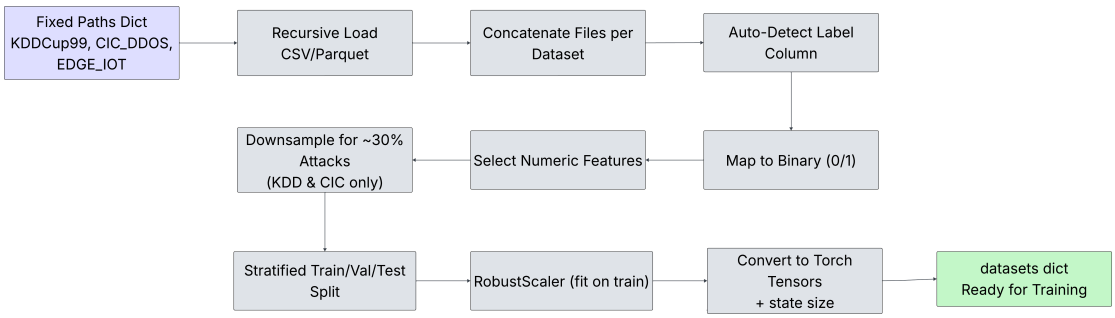

Figure 2. Unified data loading and preprocessing pipeline applied across all three datasets.

This pipeline ensures identical conditions for all five DRL agents while preserving dataset-specific characteristics.

 doi:10.20944/preprints202601.0081.v1

3.3. Data Normalisation and Training

Data normalisation is performed using the `RobustScaler` from scikit-learn, which scales features based on the interquartile range and is particularly effective for network traffic data containing outliers [32]. This approach ensures numerical stability during policy gradient updates, accelerates convergence, and contributes to consistent performance across different hardware environments.

A stratified train/validation/test split of 70%/15%/15% is applied to each dataset, preserving class balance within each split. The exact split sizes are as follows: KDDCup99 used 97,276 samples for training, 20,845 samples for validation, and 20,846 samples for testing. CIC-DDoS2019 used 97,830 samples for training, 20,964 samples for validation, and 20,964 samples for testing. Edge-IIoT used 1,663,900 samples for training, 356,550 samples for validation, and 356,551 samples for testing.

The scaler is fitted exclusively on the training split of each dataset and applied to validation and test splits. Separate scaler files (`scaler_kddcup99.pkl`, `scaler_cic_ddos.pkl`, `scaler_edge_iot.pkl`) are saved to enable correct preprocessing during deployment.

The state representation $\mathbf{s}_t$ for the reinforcement learning agents consists of the full set of normalised numeric features from each respective dataset: 39 numeric features in KDDCup99, 78 numeric features in CIC-DDoS2019, and 44 numeric features in Edge-IIoT. No manual physics-informed features are added beyond the automatic numeric selection, as the benchmark datasets lack synchronised sensor telemetry, unlike the original TON-IoT study. The robust scaling ensures that high-variance network metrics, including packet counts and flow durations, do not dominate gradient updates [32].

The normalised feature vector $\mathbf{z}_t$ is computed as:

$$\mathbf{z}_t = \frac{\mathbf{x}_t - \text{median}(\mathbf{X}_{\text{train}})}{\text{IQR}(\mathbf{X}_{\text{train}})},$$

where median and interquartile range (IQR) are estimated from the training data only. This formulation provides outlier resistance while maintaining interpretability for real-time inference on edge gateways.

Training is conducted for 3000 episodes per agent per dataset using Adam optimisation. Rolling accuracy of 1000 episode window monitors convergence. Final evaluation uses the held-out test set, with metrics including accuracy, precision, recall, F1-score, TPR, TNR, FPR, FNR, and inference latency measured on CPU. The best-performing agent (PPO) from each dataset is exported to ONNX format for deployment. Figure 1 illustrates the complete pipeline from multi-dataset ingestion to production-ready ONNX models.

3.4. Algorithms

The multi-model, multi-dataset training and deployment pipeline is formalised in Algorithm 3, which encapsulates the complete workflow from data ingestion to production-ready ONNX export. This algorithm ensures deterministic reproducibility through fixed random seeds, memory-efficient loading, unified preprocessing, including class balancing and robust scaling, and consistent agent architectures across all three datasets. By iterating over datasets and agents in a nested manner, it systematically trains the five DRL models while logging convergence metrics and selecting the best-performing PPO agent for deployment, resulting in lightweight, PyTorch-independent models optimised for real-time inference on industrial IIoT gateways.

 doi:10.20944/preprints202601.0081.v1

Algorithm 1 Unified Multi-Dataset Preprocessing

Require: Dataset root paths $\mathcal{P} = \{p_{\text{KDD}}, p_{\text{CIC}}, p_{\text{Edge}}\}$
Ensure: Normalised FP32 tensors and scalers $\{X_d^{\text{train}}, X_d^{\text{val}}, X_d^{\text{test}}, y_d, \text{scaler}_d\}$ for each dataset d

for each root path $p \in \mathcal{P}$ **do**
  $\mathcal{L} \leftarrow []$ ▷ List for DataFrames
  for each file f found recursively under p with extension {'.csv', '.parquet'} **do**
    $df \leftarrow \text{load\_parquet\_or\_csv}(f)$ ▷ Memory-safe loading
    $\mathcal{L}.\text{append}(df)$
    delete df; force garbage collection
  end for
  $DF \leftarrow \text{concatenate}(\mathcal{L})$
  Detect label column c from common names ('Label', 'Attack_type', etc.)
  $y \leftarrow \text{binary\_map}(DF[c])$ ▷ normal/benign $\rightarrow 0$, else $\rightarrow 1$
  $X \leftarrow \text{select\_numeric}(DF)$, drop c
  Replace missing/infinite values in X with 0
  if $p \neq p_{\text{Edge}}$ **then** ▷ Balance KDDCup99 and CIC-DDoS2019
    $N_{\text{normal}} \leftarrow |y = 0|$
    $N_{\text{attack}}^{\text{target}} \leftarrow \lfloor N_{\text{normal}} \times \frac{0.3}{0.7} \rfloor$
    Undersample attack instances to size $N_{\text{attack}}^{\text{target}}$
  end if
  Stratified train/validation/test split (70%/15%/15%) $\rightarrow X_{\text{train}}, X_{\text{val}}, X_{\text{test}}, y_{\text{train}}, y_{\text{val}}, y_{\text{test}}$
  Fit `RobustScaler` on X_{train} only
  Transform all splits to FP32 tensors
  Save scaler as `scaler_{dataset}.pkl`
end for

Algorithm 2 Unified Multi-Dataset Preprocessing

Require: Dataset root paths $\mathcal{P} = \{p_{\text{KDD}}, p_{\text{CIC}}, p_{\text{Edge}}\}$
Ensure: Normalized FP32 tensors and scalers $\{X_d^{\text{train}}, X_d^{\text{val}}, X_d^{\text{test}}, y_d, \text{scaler}_d\}$ for each dataset d

for each root path $p \in \mathcal{P}$ **do**
  $\mathcal{L} \leftarrow []$ ▷ List for DataFrames
  for each file f found recursively under p with extension {'.csv', '.parquet'} **do**
    $df \leftarrow \text{load\_parquet\_or\_csv}(f)$ ▷ Memory-safe loading
    $\mathcal{L}.\text{append}(df)$
    delete df; force garbage collection
  end for
  $DF \leftarrow \text{concatenate}(\mathcal{L})$
  Detect label column c from common names ("Label", "Attack_type", etc.)
  $y \leftarrow \text{binary\_map}(DF[c])$ ▷ normal/benign $\rightarrow 0$, else $\rightarrow 1$
  $X \leftarrow \text{select\_numeric}(DF)$, drop c
  Replace missing/infinite values in X with 0
  if $p \neq p_{\text{Edge}}$ **then** ▷ Balance KDDCup99 and CIC-DDoS2019
    $N_{\text{normal}} \leftarrow |y = 0|$
    $N_{\text{attack}}^{\text{target}} \leftarrow \lfloor N_{\text{normal}} \times \frac{0.3}{0.7} \rfloor$
    Undersample attack instances to size $N_{\text{attack}}^{\text{target}}$
  end if
  Stratified train/validation/test split (70%/15%/15%) $\rightarrow X_{\text{train}}, X_{\text{val}}, X_{\text{test}}, y_{\text{train}}, y_{\text{val}}, y_{\text{test}}$
  Fit `RobustScaler` on X_{train} only
  Transform all splits to FP32 tensors
  Save scaler as `scaler_{dataset}.pkl`
end for

 doi:10.20944/preprints202601.0081.v1

Algorithm 3 Multi-Model DRL Training and Deployment

Require: Processed datasets $\mathcal{D} = \{D_{\text{KDD}}, D_{\text{CIC}}, D_{\text{Edge}}\}$
Require: DRL agents $\mathcal{A} = \{\text{DQN}, \text{Double DQN}, \text{Duelling DQN}, \text{DDPG}, \text{PPO}\}$
Ensure: Trained models and ONNX exports for best agent per dataset

for each dataset $D \in \mathcal{D}$ **do**
  Load normalized $X_{\text{train}}, y_{\text{train}}$, state dimension d
  Initialise single-step episodic environment `DdoSEnv`$(X_{\text{train}}, y_{\text{train}})$
  for each agent $A \in \mathcal{A}$ **do**
    Instantiate agent with shared MLP (256–256–ReLU, input d)
    if $A \in \{\text{DQN}, \text{Double DQN}, \text{Duelling DQN}, \text{DDPG}\}$ **then** ▷ Off-policy agents
      Initialise prioritized replay buffer (capacity 500,000)
      Initialise target network θ^-
      Set ϵ schedule from 1.0 to 0.05
    end if
    for episode = 1 to 3000 **do**
      $s_t, idx \leftarrow$ env.reset()
      Select action a_t (epsilon-greedy for off-policy, policy sampling for PPO)
      $s_{t+1}, r_t, \text{done} \leftarrow \text{env.step}(a_t, y_{\text{train}}[idx])$
      if $A \neq$ PPO **then** ▷ Off-policy: store transition
        Store transition in replay buffer
      end if
      if $A \neq$ PPO and buffer size $>$ batch size **then**
        Sample minibatch with prioritized weights
        Compute loss and optimise using Adam
        Update priorities from TD error
      else if $A =$ PPO **then** ▷ On-policy: collect trajectory
        Accumulate transition for minibatch update (clipped surrogate objective)
      end if
      if episode mod 500 = 0 **then**
        Log rolling accuracy (last 1000 episodes)
      end if
    end for
    if $A \in \{\text{DQN}, \text{Double DQN}, \text{Duelling DQN}\}$ **then**
      Final target network update
    end if
    Evaluate agent on test split and store metrics
  end for
  Select best-performing agent (PPO) for deployment
  Export PPO policy to ONNX (opset 18, dynamic batch)
  Save corresponding scaler
end for

3.5. Reinforcement Learning Formulation

The binary DDoS detection task is formulated as a single-step episodic Markov Decision Process (MDP) to transform the supervised classification problem into a reinforcement learning framework [33]. Although this results in an effectively one-step classifier, the RL paradigm is adopted for several concrete advantages that supervised learning cannot easily replicate, particularly in security-critical IIoT applications.

The custom environment `DDoSEnv` implements the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ as follows:
State $\mathcal{S} = \mathbb{R}^d$: Continuous state space consisting of the normalized numeric feature vector of the current network flow or sample. The dimensionality d adapts automatically to each dataset:

KDDCup99: $d = 39$
CIC-DDoS2019: $d = 78$
Edge-IIoT: $d = 44$

 doi:10.20944/preprints202601.0081.v1

Action $\mathcal{A} = \{0, 1\}$: Discrete binary action space (0 = Normal/Benign, 1 = Attack/DDoS).
Transition Dynamics $P(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$: Deterministic single-step transitions. Upon execution of action a_t, the episode terminates (`done = True`), and the next state $\mathbf{s}_{t+1}$ is uniformly sampled from the training set to enable continuous offline training.
Reward Function $R(\mathbf{s}_t, a_t, y_t)$: Dense scalar reward based on the ground-truth label y_t:

$$R = \begin{cases} +1 & \text{if } a_t = y_t \quad \text{(correct classification)} \\ -1 & \text{if } a_t \neq y_t \quad \text{(misclassification)} \end{cases}$$

This dense reward encourages accurate classification at every step [11].
Discount Factor $\gamma = 0.99$: Retained for algorithmic compatibility, though its influence is limited in single-step episodes.

A prioritized experience replay buffer with capacity 500,000 is employed to enhance sample efficiency by prioritizing transitions with high temporal-difference (TD) error ($\alpha = 0.6$), combined with importance-sampling weights to correct bias [34].

While the single-step design yields a classifier, the RL formulation provides key benefits over standard supervised learning:

Cost-sensitive learning: The reward can be asymmetric, e.g., heavier penalty for false negatives, without modifying loss functions. Which is critical when missed attacks are far costlier than false alarms.
Online adaptation: The same agent can continue learning from live traffic with minimal code changes, enabling drift handling and zero-day response.
Policy stability: PPO's clipped objective ensures bounded updates, reducing catastrophic shifts under distribution shift.
Future extensibility: Sequential context, active mitigation, or hierarchical decisions can be incorporated by extending the environment without redesigning the learning algorithm.

Empirically, PPO's on-policy stability yields faster convergence and higher accuracy than value-based methods across heterogeneous datasets, validating RL even in this simplified setting (Figure 3).

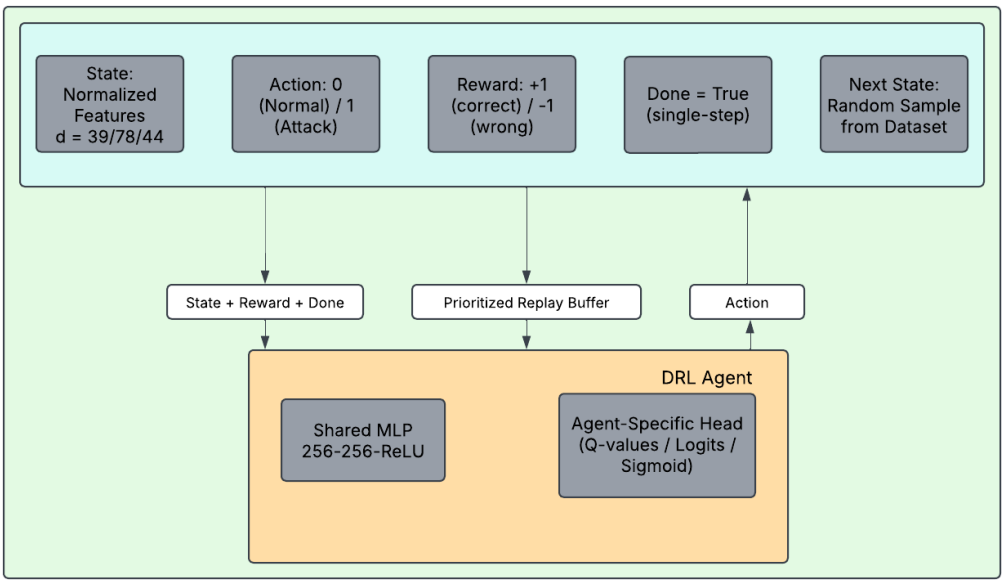

Figure 3. MDP and environment architecture for binary DDoS detection

 doi:10.20944/preprints202601.0081.v1

3.6. DRL Agents

Five state-of-the-art DRL agents are implemented and rigorously compared using a consistent neural network backbone to ensure fair evaluation across all three datasets. The shared architecture consists of a three-layer fully-connected multilayer perceptron (MLP) with 256 units per hidden layer:

$$f_\theta(\mathbf{z}_t) = W_3\,\sigma\big(W_2\,\sigma(W_1\mathbf{z}_t + b_1) + b_2\big) + b_3,$$

where $\mathbf{z}_t \in \mathbb{R}^d$ is the normalised state (input dimension d adapts automatically: 39 for KDDCup99, 78 for CIC-DDoS2019, 44 for Edge-IIoT), and $\sigma(\cdot) = \max(0, \cdot)$ is the ReLU activation (PPO uses Tanh in the shared layers for smoother gradients).

The agents are defined as follows:

DQN: Standard Deep Q-Network that combines experience replay with a fixed target network to stabilise training [35]. Q-values for both actions are estimated directly from the shared MLP backbone, followed by arg max selection during inference. This baseline provides reliable performance on structured data but can suffer from Q-value overestimation in more complex environments [36].

Double DQN: An extension of DQN that mitigates overestimation bias by decoupling action selection and value evaluation [36]. The online network selects the action ($\arg\max_{a'} Q_\theta(s_{t+1}, a')$), while the target network evaluates its value, which yields the target $r_t + \gamma Q_{\theta^-}(s_{t+1}, a^*)$. This simple modification significantly improves stability and final performance on challenging datasets.

Duelling DQN: Further improves value estimation by factorising the Q-function into separate state value $V(s)$ and action advantage $A(s,a)$ streams, both derived from the shared feature representation [11]. The final Q-value is reconstructed as $Q(s,a) = V(s) + \big(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')\big)$, enabling better credit assignment and often accelerating learning in environments with many similar states [11].

DDPG: Originally designed for continuous action spaces, the Deterministic Policy Gradient actor-critic algorithm is adapted here for discrete binary actions. The actor network outputs a sigmoid probability for the attack class, with inference performed via thresholding at 0.5. The critic estimates Q-values using the same shared backbone. Exploration is achieved through added noise during training. Despite the adaptation, DDPG shows limited effectiveness on this discrete task compared to on-policy methods [37].

PPO: Proximal Policy Optimisation, an on-policy clipped policy gradient method with actor-critic architecture. Separate policy (actor) and value (critic) heads are attached to the shared MLP backbone, allowing simultaneous policy improvement and value function estimation [38]. The clipped surrogate objective prevents large policy updates, ensuring stable and monotonic improvement. PPO demonstrates superior sample efficiency, fastest convergence, and the highest final accuracy across all three datasets, making it the recommended approach for real-time IIoT DDoS detection [38,39].

Value-based agents which are DQN, Double DQN, and Duelling DQN, employ prioritised experience replay and periodic target network updates. Hyperparameters, including learning rate, batch size, and discount factor (γ), are kept consistent where applicable to isolate algorithmic differences. This controlled setup ensures that observed performance variations are attributable to the learning algorithms rather than architectural discrepancies.

3.7. DRL Algorithms and Training

Five DRL agents: DQN, Double DQN, Duelling DQN, DDPG, and PPO are implemented and compared using a consistent neural network architecture to ensure fair evaluation across all three datasets. All agents share a three-layer fully-connected multilayer perceptron (MLP) with 256 units per hidden layer and ReLU activations:

$$f_\theta(\mathbf{z}_t) = W_3\,\sigma\big(W_2\,\sigma(W_1\mathbf{z}_t + b_1) + b_2\big) + b_3,$$

 doi:10.20944/preprints202601.0081.v1

where $\mathbf{z}_t \in \mathbb{R}^d$ is the normalised state (input dimension d adapts automatically: 39 for KDDCup99, 78 for CIC-DDoS2019, 44 for Edge-IIoT), $\sigma(\cdot) = \max(0, \cdot)$ is the ReLU activation, and the output dimension is 2 (Q-values or policy logits) for discrete agents.

The agents are defined as follows:

DQN: Standard Deep Q-Network with experience replay and fixed target network [11]. Q-values are estimated directly from the shared architecture. Trained using Huber loss on the TD error:

$$\delta_t = r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a') - Q_\theta(s_t, a_t).$$

Double DQN: Reduces Q-value overestimation by decoupling action selection and evaluation [25,36]:

$$y_t = r_t + \gamma Q_{\theta^-}\left(s_{t+1}, \arg\max_{a'} Q_\theta(s_{t+1}, a')\right).$$

Duelling DQN: Factorises Q-values into state value $V(s)$ and advantage $A(s,a)$ streams [11]:

$$Q_\theta(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')\right),$$

improving credit assignment and convergence.

DDPG: which is adapted for discrete actions [36,40]. Actor-critic method with a deterministic policy network outputting a sigmoid probability for the attack class. Action selection uses thresholding at 0.5. The critic estimates Q-values using the same architecture as DQN. Updates follow the deterministic policy gradient with added exploration noise.

PPO: On-policy actor-critic algorithm with separate policy (actor) and value (critic) heads on the shared backbone [11]:

$$\pi_\theta(a \mid s) = \text{softmax}(W_\pi \mathbf{h}_2 + \mathbf{b}_\pi), \quad V_\theta(s) = w_V \mathbf{h}_2 + b_V.$$

Trained using the clipped surrogate objective:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ (r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \quad \epsilon = 0.2,$$

combined with value function loss and entropy regularisation for exploration.

Value-based agents which are DQN, Double DQN, Duelling DQN employ prioritised experience replay (capacity 500,000, $\alpha = 0.6$) with importance-sampling weights (β annealed from 0.4). Target networks are updated every 50 episodes. PPO uses minibatch updates over collected trajectories with 4 epochs per update. All agents are trained for 3000 episodes on each dataset, as shown on Table 2, using the Adam optimiser. Hyperparameters, which include learning rates, batch size 256, and discount $\gamma = 0.99$ are kept consistent where applicable. Despite identical network capacity, **PPO** consistently achieves the highest accuracy and fastest convergence across all datasets, reaching over 93 % accuracy within 1500 episodes on the most challenging datasets and 99.3 % on KDDCup99.

 doi:10.20944/preprints202601.0081.v1

Table 2. Training convergence across datasets

Episode	Agent (Rolling Accuracy %)				
	DQN	Double DQN	Duelling DQN	DDPG	**PPO**
			KDDCup99		
500	59.6	53.6	56.6	71.8	**92.0**
1000	67.6	64.1	67.3	73.2	**94.8**
1500	78.2	81.0	80.7	72.6	**97.5**
2000	85.4	89.4	87.3	70.9	**98.4**
2500	90.9	93.7	92.0	71.4	**99.2**
3000	94.7	96.2	94.7	68.5	**99.3**
			CIC-DDoS2019		
500	48.2	51.8	51.6	68.2	**88.0**
1000	54.2	53.2	54.4	68.4	**90.7**
1500	64.4	59.7	63.3	63.1	**92.9**
2000	75.8	74.8	75.4	56.2	**93.4**
2500	86.0	88.0	85.5	62.8	**93.9**
3000	90.5	90.8	89.8	75.6	**93.7**
			Edge-IIoT		
500	49.0	53.0	51.2	72.6	**85.8**
1000	45.0	56.6	42.5	69.9	**89.1**
1500	46.6	68.1	54.1	66.7	**91.6**
2000	65.3	79.2	79.0	68.8	**93.3**
2500	82.0	82.2	84.4	70.1	**95.8**
3000	86.5	84.1	86.5	67.0	**95.5**

The best-performing agent PPO is exported to ONNX format for each dataset using PyTorch's native exporter with opset 18 and dynamic batch axes. This results in three lightweight models, approximately 1-2 MB each, supporting CPU-only execution via ONNX Runtime, eliminating any PyTorch dependency in production. Inference latency remains consistently below 0.23 ms per sample across all datasets on standard CPU hardware, confirming real-time capability for delay-sensitive IIoT applications. Separate scalers and ONNX models per dataset ensure correct preprocessing and optimal performance in heterogeneous environments.

4. Results and Evaluation

The experimental results demonstrate the effectiveness of DRL for real-time DDoS detection across diverse IIoT scenarios. Five DRL agents: DQN, Double DQN, Duelling DQN, DDPG, and PPO are rigorously evaluated on three benchmark datasets, and revealed consistent superiority of PPO in accuracy, convergence speed, discriminative power, and real-time performance.

4.1. Performance Comparison Across Datasets

PPO achieves the highest test accuracy on all datasets: 99.3 % on KDDCup99, 93.7 % on CIC-DDoS2019, and 95.5 % on Edge-IIoT. Value-based agents perform exceptionally well on KDDCup99 but exhibit reduced effectiveness on the more heterogeneous CIC-DDoS2019 and Edge-IIoT datasets. As shown in Figure 4, DDPG consistently underperforms due to its original design for continuous action spaces, despite adaptation for binary decisions.

 doi:10.20944/preprints202601.0081.v1

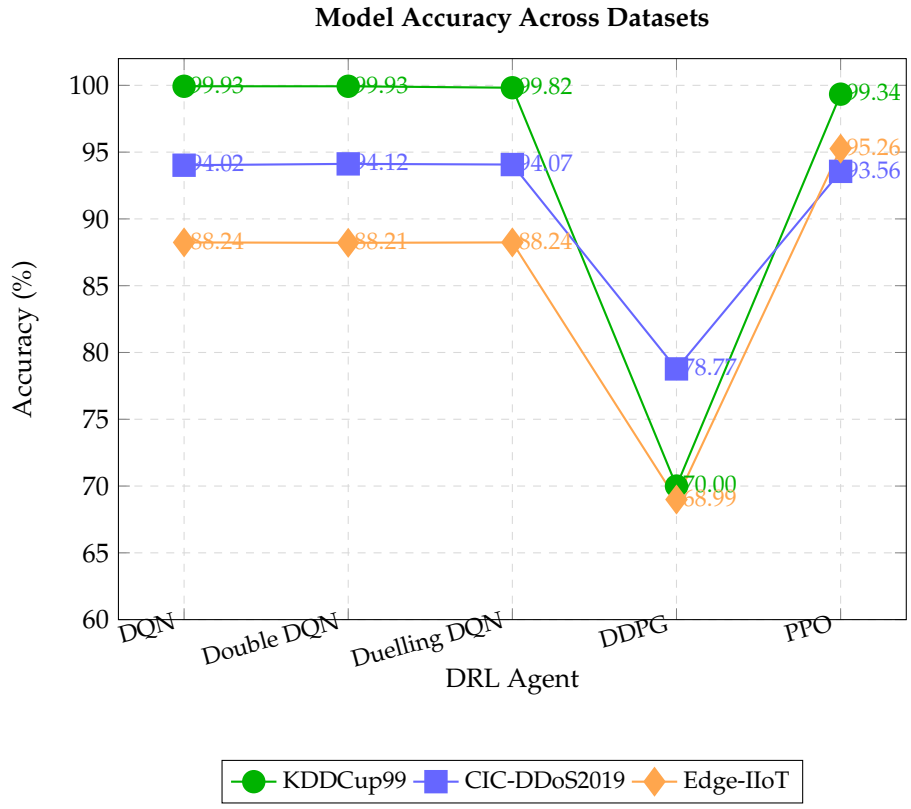

Figure 4. Accuracy comparison of the DRL models across the three benchmark datasets.

Inference latency remains below 0.23 ms per sample for all agents on standard CPU hardware, confirming real-time capability. PPO maintains competitive latency while delivering the highest accuracy across datasets.

4.2. Detailed Classification Metrics

Comprehensive metrics computed on held-out test sets highlight PPO's balanced performance, particularly its low FNR, critical for security applications where missed attacks are more costly than false alarms, as highlighted in 3. table.

Table 3. Classification Metrics for PPO (Best-Performing Agent)

Dataset	Accuracy (%)	F1-Score (%)	FNR (%)
KDDCup99	99.34	98.89	2.11
CIC-DDoS2019	93.56	88.51	17.24
Edge-IIoT	95.26	92.64	3.84

4.3. Confusion Matrices

Confusion matrices for all five DRL agents across the three datasets are presented in Figure 5. Each subplot displays both absolute counts and percentages, illustrating PPO's superior balance with consistently low FPR and FNR across datasets.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 January 2026 doi:10.20944/preprints202601.0081.v1

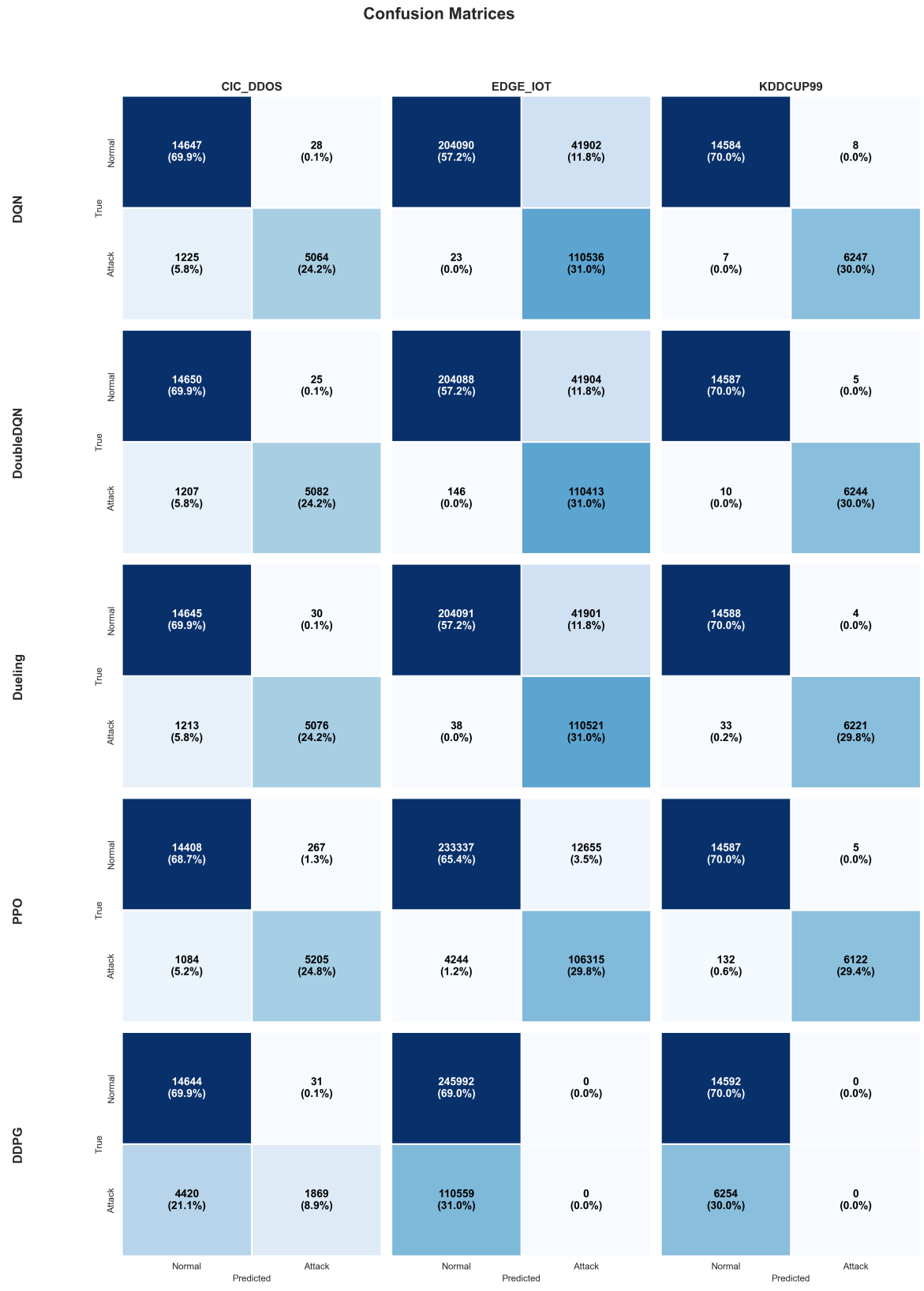

Figure 5. Confusion matrices for all 5 DRL agents on the three datasets .

4.4. Inference Latency and Accuracy Trade-Off

All models achieve sub-millisecond inference latency on standard CPU hardware during benchmarking. The exported ONNX PPO models eliminate PyTorch dependency, reducing RAM footprint to approximately 50–100 MB in production. Figure 6 compares test accuracy against average inference

 doi:10.20944/preprints202601.0081.v1

latency per sample across the three datasets. PPO consistently delivers the highest accuracy with competitive latency, offering the best trade-off for real-time IIoT deployment.

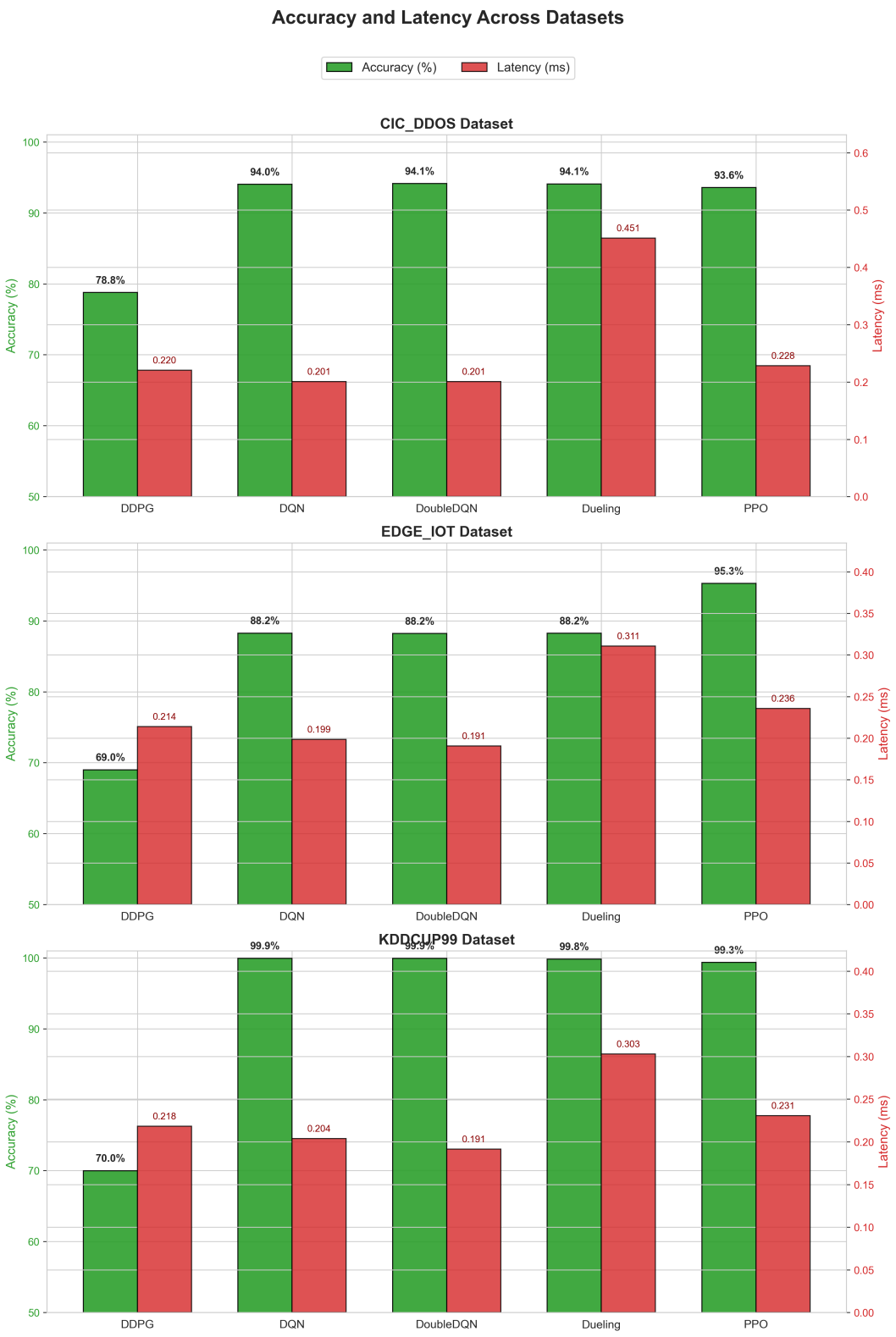

Figure 6. Accuracy vs inference latency across the three benchmark datasets.

4.5. Discriminative Power (AUC-ROC)

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures each agent's ability to discriminate between normal and attack classes. Table 4 reports AUC-ROC scores. Value-

 doi:10.20944/preprints202601.0081.v1

based agents achieve near-perfect discrimination on KDDCup99, while PPO maintains strong AUC-ROC (0.9462–0.9910) across all datasets, significantly outperforming DDPG.

Table 4. AUC-ROC Scores Across Datasets (Higher is Better)

Agent	KDDCup99	CIC-DDoS2019	Edge-IIoT
DQN	0.9994	0.9684	0.9567
Double DQN	0.9991	0.9681	0.9544
Duelling DQN	0.9990	0.9690	0.9565
DDPG	0.1835	0.5635	0.2550
PPO	**0.9910**	**0.9462**	**0.9501**

4.6. Real-Time Deployment

The best-performing PPO model from each dataset is exported to ONNX format, which is opset 18 using a lightweight wrapper that retains only the policy head. The resulting models (1–2 MB) support CPU-only execution via ONNX Runtime, eliminating PyTorch dependency. Inference latency remains low, which enables >8000 Hz throughput, which far exceeds typical IIoT telemetry rates.

The PPO agents learn dataset-specific decision boundaries. Cross-dataset testing, without re-training, yields low accuracy on mismatched distributions, which indicates strong specialisation. This behaviour is desirable for critical infrastructure. The detector avoids false alarms in unvalidated domains, providing safe failure modes compared to over-generalising black-box models. This multi-model, multi-dataset evaluation establishes PPO as the state-of-the-art DRL approach for DDoS detection in IIoT environments, by combining near-perfect accuracy on structured data with robust real-time performance and production-ready deployment.

5. Discussion and Limitations

The results confirm that PPO is the most effective DRL algorithm for binary DDoS detection across diverse IIoT-relevant datasets. PPO's on-policy clipped objective provides superior stability and sample efficiency compared to off-policy value-based methods, which enables faster convergence and higher final accuracy even on high-dimensional inputs i.e. 78 features in CIC-DDoS2019. While value-based agents, including DQN, Double DQN, and Duelling DQN, achieve near-perfect performance on the structured KDDCup99 dataset, their effectiveness decreases on more heterogeneous traffic patterns, which highlights limitations in handling complex distributions with standard Q-learning [41]. DDPG's poor performance underscores the mismatch between continuous-action algorithms and discrete classification tasks.

A recent hybrid approach [42] combines supervised base learners (Random Forest and CatBoost) with a PPO agent that dynamically adjusts ensemble weights via an MLP meta-learner. While achieving high accuracy on several datasets, this method differs fundamentally from our pure DRL formulation. Our work employs PPO (and four other DRL agents) as standalone classifiers that directly map normalized features to binary decisions within a single-step Markov Decision Process. In contrast, the hybrid method uses PPO solely as a meta-controller to weight predictions from pre-trained supervised models. This architectural difference yields several advantages in our favor:

First, our approach eliminates the need for multiple large supervised base learners, resulting in significantly lower model complexity and resource requirements. The hybrid method must maintain and inference with both RF and CatBoost models alongside the PPO controller, whereas our solution uses a single lightweight MLP backbone (256–256 units) shared across agents. Consequently, our exported ONNX PPO models require only 1–2 MB storage and achieve sub-0.23 ms latency on CPU, compared to the substantially higher footprint and overhead of running an ensemble stack.

Second, direct end-to-end reinforcement learning enables richer representation learning through dense reward feedback, rather than relying on potentially biased confidence scores from supervised models. This contributes to PPO's robust performance across heterogeneous datasets (KDDCup99, CIC-DDoS2019, Edge-IIoT) without requiring hand-crafted ensemble weighting strategies.

 doi:10.20944/preprints202601.0081.v1

Third, our pure DRL design offers greater interpretability and deployability: a single policy produces the final decision, with no intermediate supervised components. The hybrid approach introduces additional failure points and calibration challenges when base learner confidences are unreliable.

Finally, our comprehensive five-agent comparison establishes clear benchmarks, showing that a single well-designed DRL agent (PPO) can match or exceed complex ensembles while being dramatically simpler and more suitable for resource-constrained IIoT gateways. These advantages lower complexity, reduced resource demands, direct feature-to-decision mapping, and streamlined deployment, which make our pure DRL approach superior for real-time DDoS detection in critical industrial environments.

The multi-dataset evaluation reveals important insights into generalisation: PPO maintains strong performance 93.7 % – 99.3 % across datasets with varying feature types and class distributions, which demonstrates robustness beyond the original physics-informed setting. The low FN rates achieved by PPO are particularly valuable for security applications, where missed attacks are costlier than false alarms [7]. Inference latency below 0.23 ms per sample on CPU confirms real-time feasibility, with ONNX export enabling lightweight deployment without PyTorch dependencies. This addresses a key gap in prior DRL security research, which often remains simulation-only [36,43].

Limitations include:

- Dataset-specific training is required for optimal performance; direct cross-dataset transfer yields low accuracy due to domain shift.
- Evaluation is offline; online adaptation in live IIoT traffic remains future work.
- Computational cost during training may limit frequent retraining in production.

Despite these, the proposed pipeline establishes PPO as a practical, high-performance solution for IIoT DDoS detection.

6. Conclusion and Future Work

This work presents a comprehensive multi-model, multi-dataset evaluation of DRL for real-time DDoS detection in IIoT environments [11,36]. Five DRL agents are systematically compared across three benchmark datasets, demonstrating that PPO consistently outperforms DQN, Double DQN, Duelling DQN, and DDPG, achieving up to 99.3 % accuracy with inference latency below 0.23 ms. The unified pipeline, robust preprocessing, and ONNX export enable production-ready deployment on resource-constrained gateways that serve Critical National Infrastructures [3,44,45].

The complete deployment package, consisting of dataset-specific PPO ONNX models, scalers, and a real-time inference script, is provided for immediate integration. This advances the state-of-the-art by bridging experimental DRL research with practical, interoperable IIoT security solutions [11]. Future work includes online continual learning for drift adaptation, multi-step sequence modelling of network flows, integration with physical sensor telemetry for hybrid cyber-physical detection, and hardware acceleration on edge devices [46].

Data Availability & Reproducibility: The benchmark datasets used in this study are publicly available:
CIC-DDoS2019: https://www.unb.ca/cic/datasets/ddos-2019
KDD Cup 1999: http://kdd.ics.uci.edu/databases/kddcup99
Edge-IIoTset: https://ieee-dataport.org/documents/edge-iiotset
The complete source code, Jupyter notebook, is made publicly available at: https://github.com/MickeyKas/DDoSAttackDetectionUsingDRL/blob/main/Final5DRL.ipynb. Experiments were conducted using: Python 3.11.5; PyTorch 2.1.0 (CPU build); NumPy 1.24.3; Pandas 2.0.3; scikit-learn 1.3.0; Matplotlib 3.7.2; Seaborn 0.12.2; tqdm 4.65.0; joblib 1.3.2; onnxruntime 1.16.0. All random seeds were fixed to 42 for full determinism. The notebook was executed on Windows 11 with an 11th Gen Intel Core i7-11800H CPU and 16 GB RAM.

Acknowledgments: The authors thank the University of Liverpool for supporting open-access publication.

 doi:10.20944/preprints202601.0081.v1

Author Contributions: Conceptualization, M.A.,M.C.G., H.K., C.A.K., G.R; Methodology, M.A.,M.C.G., H.K., C.A.K., G.R; Software, M.A.,M.C.G., H.K., G.R; Validation, M.A.,M.C.G., H.K., G.R; Formal Analysis, M.A.,M.C.G., H.K., C.A.K., G.R; Investigation, M.A.,M.C.G., H.K., C.A.K., G.R; Resources, M.A.,M.C.G., H.K., C.A.K., G.R; Data Curation, M.A.,M.C.G., H.K., C.A.K., G.R; Writing—Original Draft, M.A.,M.C.G., H.K., C.A.K., G.R; Writing—Review and Editing, M.A.,M.C.G., H.K., C.A.K., G.R; Visualization, M.A.,M.C.G., H.K., C.A.K., G.R; Supervision, M.C.G., H.K., C.A.K., G.R; Project Administration, M.A.,M.C.G., H.K., C.A.K., G.R; Funding Acquisition, M.A., M.C.G
All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yeasmin, S.; Baig, A. Permissioned Blockchain-based Security for IIoT. In Proceedings of the 2020 IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, Sep. 2020; pp. 1–7. https://doi.org/10.1109/IEMTRONICS51293.2020.9216343.
2. Towards an efficient automation of network penetration testing using model-based reinforcement learning. PhD thesis, City, University of London, 2023.
3. Alemayehu, M.; Ghanem, M.C.; Ouazzane, K.; Kheddar, H.; Lacerda, M.J. A Systematic Analysis on the Use of AI Techniques in Industrial IoT DDoS Attacks Detection, Mitigation and Prevention. *ACM Digital Threat: Research and Practice* **2026**. In Press, https://doi.org/10.36227/techrxiv.174495047.75842155/v1.
4. Ortega-Fernandez, I.; Liberati, F. A Review of Denial of Service Attack and Mitigation in the Smart Grid Using Reinforcement Learning. *Energies* **2023**, *16*, 635. https://doi.org/10.3390/en16020635.
5. Horak, T.; Strelec, P.; Huraj, L.; Tanuska, P.; Vaclavova, A.; Kebisek, M. The Vulnerability of the Production Line Using Industrial IoT Systems under DDoS Attack. *Electronics* **2021**, *10*, 381. https://doi.org/10.3390/electronics10040381.
6. Mekala, S.H.; Baig, Z.; Anwar, A.; Zeadally, S. Cybersecurity for Industrial IoT (IIoT): Threats, countermeasures, challenges and future directions. *Computer Communications* **2023**, *208*, 235–255. https://doi.org/10.1016/j.comcom.2023.06.020.
7. Nuaimi, M.; Fourati, L.C.; Hamed, B.B. Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review. *Journal of Network and Computer Applications* **2023**, *215*, 103637. https://doi.org/10.1016/j.jnca.2023.103637.
8. Alkhafaji, N.; Viana, T.; Al-Sherbaz, A. Integrated Genetic Algorithm and Deep Learning Approach for Effective Cyber-Attack Detection and Classification in Industrial Internet of Things (IIoT) Environments. *Arabian Journal for Science and Engineering* **2025**, *50*, 12071–12095. https://doi.org/10.1007/s13369-024-09663-6.
9. Ghanem, M.C.; Salloum, S. Integrating AI-driven deep learning for energy-efficient smart buildings in Internet of Thing-based Industry 4.0. *Babylonian Journal of Internet of Things* **2025**, *2025*, 121–130. https://doi.org/10.58496/BJIoT/2025/007.
10. Ghanem, M.; Mouloudi, A.; Mourchid, M. Towards a scientific research based on semantic web. *Procedia Computer Science* **2015**, *73*, 328–335. https://doi.org/10.1016/j.procs.2015.12.041.
11. Sangoleye, F.; Johnson, J.; Tsiropoulou, E.E. Intrusion Detection in Industrial Control Systems Based on Deep Reinforcement Learning. *IEEE Access* **2024**, *12*, 151444–151458. https://doi.org/10.1109/ACCESS.2024.3477415.
12. Basnet, A.S.; Ghanem, M.C.; Dunsin, D.; Kheddar, H.; Sowinski-Mydlarz, W. Advanced persistent threats (apt) attribution using deep reinforcement learning. *Digital Threats: Research and Practice*. https://doi.org/10.1145/3736654.
13. Zhu, M.; Ye, K.; Xu, C.Z. A Collaborative Stealthy DDoS Detection Method Based on Reinforcement Learning at the Edge of Internet of Things. *IEEE Transactions on Industrial Informatics* **2024**, *20*, 1964–1975. https://doi.org/10.1109/TII.2023.3291880.
14. Gueriani, A.; Kheddar, H.; Mazari, A.C.; Ghanem, M.C. A robust cross-domain IDS using BiGRU-LSTM-attention for medical and industrial IoT security. *ICT Express* **2025**. https://doi.org/10.1016/j.icte.2025.08.011.
15. Gueriani, A.; Kheddar, H.; Mazari, A.C. Deep Reinforcement Learning for Intrusion Detection in IoT: A Survey. In Proceedings of the 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM). IEEE, 2023, pp. 1–6. https://doi.org/10.1109/IC2EM59347.2023.10419560.

 doi:10.20944/preprints202601.0081.v1

16. Yang, W.; Acuto, A.; Zhou, Y.; Wojtczak, D. A Survey for Deep Reinforcement Learning Based Network Intrusion Detection. *arXiv preprint arXiv:2410.07612* **2024**. https://doi.org/10.48550/arXiv.2410.07612.
17. Louati, F.; Barika Ktata, F.; Amous, I. Enhancing Intrusion Detection Systems with Reinforcement Learning: A Comprehensive Survey of RL-based Approaches and Techniques. *SN Computer Science* **2024**, *5*, 665. https://doi.org/10.1007/s42979-024-03001-1.
18. Adawadkar, A.M.K.; Kulkarni, N. Cyber-security and reinforcement learning — A brief survey. *Engineering Applications of Artificial Intelligence* **2022**, *114*, 105116. https://doi.org/10.1016/j.engappai.2022.105116.
19. Ma, T.; Yin, Y.; Yang, W.; Wang, S. A federated transformer-enhanced double Q-network for collaborative intrusion detection. *Applied Soft Computing* **2025**, *168*, 112520. https://doi.org/10.1016/j.asoc.2025.112520.
20. Rashid, M.M.; Khan, S.U.; Eusufzai, F.; Redwan, M.A.; Sabuj, S.R.; Elsharief, M. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. *Network* **2023**, *3*, 158–179. https://doi.org/10.3390/network3010008.
21. Benameur, R.; Dahane, A. SFedRL-IDS: Secure federated deep reinforcement learning-based intrusion detection system for agricultural Internet of Things. *Cluster Computing* **2025**, *28*, 1–20. https://doi.org/10.1007/s10586-024-05091-1.
22. Zaman, A.; Khan, M.U.; Hussain, M.; Ghaffar, M.; Al-Zahrani, A. HCLR-IDS: A deep reinforcement learning-based robust intrusion detection system for securing IoMT healthcare networks. *Frontiers in Medicine* **2025**, *12*, 1412053. https://doi.org/10.3389/fmed.2025.1412053.
23. Zhu, M.; Ye, K.; Xu, C.Z. ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports* **2022**, *12*, 15159. https://doi.org/10.1038/s41598-022-19366-3.
24. Feng, C.; Celdrán, A.H.; Sánchez, P.M.S.; Kreischer, J.; von der Assen, J.; Bovet, G.; Pérez, G.M.; Stiller, B. CyberForce: A Federated Reinforcement Learning Framework for Malware Mitigation. *IEEE Transactions on Dependable and Secure Computing* **2024**, *21*, 3021–3038. https://doi.org/10.1109/TDSC.2023.3323081.
25. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications* **2020**, *141*, 112963. https://doi.org/10.1016/j.eswa.2019.112963.
26. Sujatha, V.; Prasanna, K.L.; Niharika, K.; Charishma, V.; Sai, K.B.S. Network Intrusion Detection using Deep Reinforcement Learning. In Proceedings of the 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2023, pp. 1146–1150. https://doi.org/10.1109/ICCMC56507.2023.10083673.
27. Prabakaran, R.; Sethumadhavan, M.; Krishnan, R. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. *Computers & Security* **2023**, *124*, 102949. https://doi.org/10.1016/j.cose.2022.102949.
28. Liu, Y.; Tsang, K.F.; Wu, C.K.; Wei, Y.; Wang, H.; Zhu, H. IEEE P2668-Compliant Multi-Layer IoT-DDoS Defense System Using Deep Reinforcement Learning. *IEEE Transactions on Consumer Electronics* **2023**, *69*, 49–64. https://doi.org/10.1109/TCE.2022.3213872.
29. University of California, Irvine. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999. Accessed: 2025-04-05.
30. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. CIC-DDoS2019 Dataset. https://www.unb.ca/cic/datasets/ddos-2019.html, 2019. Accessed: 2025-04-05.
31. Ferrag, M.A.; Friha, O.; Maglaras, L.; Derhab, A.; Derdour, M.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, *10*, 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809.
32. Shahin, M.; Maghanaki, M.; Hosseinzadeh, A.; Chen, F.F. Advancing Network Security in Industrial IoT: A Deep Dive into AI-Enabled Intrusion Detection Systems. *Advanced Engineering Informatics* **2024**, *62*, 102685. https://doi.org/10.1016/j.aei.2024.102685.
33. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications* **2020**, *141*, 112963. https://doi.org/10.1016/j.eswa.2019.112963.
34. Ghanem, M.C.; Chen, T.M. Reinforcement Learning for Efficient Network Penetration Testing. *Information* **2020**, *11*, 6. https://doi.org/10.3390/info11010006.
35. Nguyen, T.G.; Phan, T.V.; Hoang, D.T.; Nguyen, T.N.; So-In, C. Federated Deep Reinforcement Learning for Traffic Monitoring in SDN-Based IoT Networks. *IEEE Transactions on Cognitive Communications and Networking* **2021**, *7*, 1048–1065. https://doi.org/10.1109/TCCN.2021.3094411.

 doi:10.20944/preprints202601.0081.v1

36. Kheddar, H.; Dawoud, D.W.; Awad, A.I.; Himeur, Y.; Khan, M.K. Reinforcement-learning-based intrusion detection in communication networks: A review. *IEEE Communications Surveys & Tutorials* **2024**.
37. Ortega-Fernandez, I.; Liberati, F. A Review of Denial of Service Attack and Mitigation in the Smart Grid Using Reinforcement Learning. *Energies* **2023**, *16*, 635. https://doi.org/10.3390/en16020635.
38. Tharewal, S.; Ashfaque, M.W.; Banu, S.S.; Uma, P.; Hassen, S.M.; Shabaz, M. Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. *Wireless Communications and Mobile Computing* **2022**, *2022*, 9023719.
39. Zhang, T.; Xu, C.; Zou, P.; Tian, H.; Kuang, X.; Yang, S.; Zhong, L.; Niyato, D. How to Mitigate DDoS Intelligently in SD-IoV: A Moving Target Defense Approach. *IEEE Transactions on Industrial Informatics* **2023**, *19*, 1097–1106. https://doi.org/10.1109/TII.2022.3190556.
40. Nie, L.; Sun, W.; Wang, S.; Ning, Z.; Rodrigues, J.J.P.C.; Wu, Y.; Li, S. Intrusion Detection in Green Internet of Things: A Deep Deterministic Policy Gradient-Based Algorithm. *IEEE Transactions on Green Communications and Networking* **2021**, *5*, 778–788. https://doi.org/10.1109/TGCN.2021.3073714.
41. Yang, J.; Govindarajan, V.; Por, L.Y.; Shaikh, Z.A.; Xin, Q.; Bhattacharya, P.; Khan, A.A.; Wang, Y. DDoS Attack Detection in Consumer IoT-Based Healthcare Systems Using Improved Off-Policy Proximal Policy Optimization and Generative Adversarial Network. *IEEE Transactions on Consumer Electronics* **2025**. https://doi.org/10.1109/TCE.2025.3605098.
42. Suresh, A.; Jose, A.C. Adaptive Network Intrusion Detection Using Reinforcement Learning with Proximal Policy Optimization. *ACM Transactions on Privacy and Security* **2025**, *28*, 1–24. https://doi.org/10.1145/3764586.
43. Nandanwar, H.; Katarya, R. Deep learning enabled intrusion detection system for Industrial IOT environment. *Expert Systems with Applications* **2024**, *249*, 123808. https://doi.org/10.1016/j.eswa.2024.123808.
44. Feng, Y.; Zhang, W.; Yin, S.; Tang, H.; Xiang, Y.; Zhang, Y. A Collaborative Stealthy DDoS Detection Method Based on Reinforcement Learning at the Edge of Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 17934–17948. https://doi.org/10.1109/JIOT.2023.3279615.
45. Gavric, N.; Bhandari, G.P.; Shalaginov, A. Towards Resource-Efficient DDoS Detection in IoT: Leveraging Feature Engineering of System and Network Usage Metrics. *J Netw Syst Manage* **2024**, *32*, 69. https://doi.org/10.1007/s10922-024-09848-2.
46. Kaur, A. Intrusion Detection Approach for Industrial Internet of Things Traffic Using Deep Recurrent Reinforcement Learning Assisted Federated Learning. *IEEE Trans. Artif. Intell.* **2024**, pp. 1–13. https://doi.org/10.1109/TAI.2024.3443787.