



Explainable Hybrid Intelligence for Autonomous Cyber Defense: Adaptive Intrusion Detection and Real-time Firewall Enforcement Using Reinforcement Learning

Magi Hossameldin Mahfouz¹Raghda Essam Ali^{1*}Ayat Mahmoud²¹*School of Computing & Digital Tech., ESLSCA University, Egypt*²*October University for Modern Sciences & Arts, Egypt** Corresponding author's Email: raghda.essam@eslsca.edu.eg

Abstract: Conventional rule-based firewalls rely on static signatures, limiting adaptability to unseen attacks, while AI-enabled commercial tools remain proprietary and lack transparency. This paper presents Net-Knight, an explainable hybrid intelligence framework integrating supervised intrusion detection, unsupervised anomaly detection, reinforcement learning (RL)-based mitigation, and Linux nftables enforcement. A LightGBM classifier trained on 22 engineered NetFlow features from NF-UQ-NIDS-v2 performs intrusion detection, while an autoencoder ensemble supplies complementary anomaly evidence. These signals form a 28-dimensional state used by a MaskablePPO agent to select among five mitigation actions. The IDS achieves 96.82% accuracy and a 0.9681 macro F1-score across nine classes; the RL agent attains a 96.9% rule-match rate; and the integrated decision pipeline reaches 96.3% end-to-end decision accuracy at 8.7 ms mean processing latency in the controlled evaluation. A proposed Hierarchical Confusion-Aware Disambiguation (HCAD) extension is evaluated separately through Monte Carlo sensitivity analysis; its projected gains are not treated as experimentally measured performance.

Keywords: Intrusion detection systems, Reinforcement learning, Explainable artificial intelligence, Autonomous cyber defence, Network security, Anomaly detection, LightGBM, MaskablePPO, Nftables.

1. Introduction

The increasing scale and sophistication of cyber threats have placed network security at the forefront of modern digital infrastructure protection. Organisations are continuously exposed to heterogeneous attacks ranging from reconnaissance and brute-force attempts to distributed denial-of-service (DDoS), malware, and application-layer exploits [1, 2]. Conventional firewalls remain an essential first line of defence; however, their effectiveness is largely determined by predefined rules, signatures, and administrator-defined policies. This dependence on prior knowledge makes static enforcement less effective when confronted with previously unseen or behaviourally evolving attacks that do not conform to known indicators of compromise.

Machine Learning (ML)-based Intrusion Detection Systems (IDSs) offer a more adaptive alternative by learning discriminative patterns directly from network traffic. Recent studies have demonstrated that ensemble and tree-based learning models can achieve effective intrusion classification, while explainability techniques such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) can provide additional insight into the features driving individual predictions [3]. Nevertheless, accurate classification alone does not constitute autonomous defence. Most IDS architectures terminate their operation after generating an alert, leaving the subsequent mitigation decision to a human administrator. This separation between detection and enforcement creates a critical response gap, particularly for high-velocity attacks in which

malicious activity can escalate before manual intervention becomes effective.

Reinforcement Learning (RL) provides a suitable mechanism for bridging this gap because it formulates cyber defence as a sequential decision problem rather than solely as a classification task. Recent hybrid IDS frameworks have consequently begun combining supervised or deep-learning-based detection with anomaly modelling and RL-driven adaptation, demonstrating the potential of intelligent response mechanisms alongside conventional traffic classification [4, 5]. An RL agent can learn a mitigation policy that maps the observed security state to an appropriate defensive action according to its expected consequences. Recent studies have demonstrated the potential of RL for adaptive intrusion detection and self-adjusting cyber-defence mechanisms [4, 6]. However, several challenges remain before such approaches can support practical autonomous firewall operation. In particular, detection uncertainty must be incorporated into the decision process, invalid or unnecessarily disruptive responses must be constrained, learned decisions must be translated into executable firewall actions, and autonomous mitigation must remain interpretable to network administrators.

These requirements are particularly relevant for small and medium-sized enterprises (SMEs), where limited security expertise and operational resources can restrict continuous human monitoring [7]. They are also important in larger environments, where high alert volumes and repeated manual policy updates increase administrative overhead. Recent research has also emphasized lightweight and interpretable IDS architectures as a means of balancing detection effectiveness with computational and operational constraints [3, 8].

To address these challenges, this paper presents Net-Knight, an explainable hybrid intelligence framework for adaptive intrusion detection and real-time firewall enforcement. Hybrid intrusion-detection architectures have further shown that combining supervised classification with unsupervised anomaly modelling can exploit complementary evidence from known attack patterns and deviations from normal traffic behaviour [5, 9]. The supervised component identifies known traffic classes, whereas the anomaly component provides complementary evidence of behavioural deviation. These signals, together with traffic-state information, form the observation space of a Maskable Proximal Policy Optimization (MaskablePPO) agent [10, 11], which selects an admissible mitigation action according to the observed network state. The selected action is subsequently translated into an executable

Linux nftables rule [12], thereby closing the loop between traffic analysis, defensive decision-making, and firewall enforcement.

Explainability is incorporated into this process to associate enforcement decisions with the evidence that influenced them, while a semi-automatic operating mode preserves human approval when fully autonomous response is inappropriate. Furthermore, analysis of the base IDS reveals a localized source of residual error among closely related web-layer attack classes. Rather than increasing model complexity globally, this work introduces Hierarchical Confusion-Aware Disambiguation (HCAD), a targeted second-stage mechanism designed specifically for these ambiguous predictions. Its expected contribution is evaluated separately through a confusion-matrix-grounded Monte Carlo simulation, ensuring that simulated improvements are distinguished from experimentally measured performance.

The main contributions of this work are summarized as follows:

- A class-balanced, feature-engineered multi-class IDS based on the NF-UQ-NIDS-v2 benchmark [13, 14], including a comparative assessment of LightGBM, XGBoost [15], and Random Forest across nine traffic classes.
- An adaptive mitigation mechanism based on MaskablePPO [10, 11] that jointly exploits IDS predictions, anomaly evidence, and traffic-pressure information to select context-dependent defensive actions, with evaluation against random and heuristic policies and across multiple training seeds.
- A closed-loop implementation that converts learned mitigation decisions into real-time Linux nftables enforcement [12], complemented by decision-level explanations and a human-in-the-loop approval mode.
- HCAD, a targeted hierarchical classification strategy designed from the empirically observed confusion patterns of the primary IDS, together with a Monte Carlo assessment of its expected performance improvement.
- An end-to-end evaluation of Net-Knight in terms of intrusion detection performance, mitigation decision accuracy, critical error rate, and per-flow processing latency, providing system-level evidence of its suitability for real-time network defence.

Unlike approaches that consider intrusion detection, mitigation policy learning, or firewall enforcement in isolation, Net-Knight treats them as components of a unified defensive control loop. The

resulting framework is intended to combine the predictive capability of data-driven intrusion detection with adaptive RL-based response, interpretable decision support, and enforceable network-level mitigation.

The remainder of this paper is organized as follows. Section 2 reviews related work on AI-assisted intrusion detection, reinforcement-learning-based network defence, and explainable security systems. Section 3 describes the dataset, preprocessing procedures, Net-Knight architecture, learning models, and the proposed HCAD mechanism. Section 4 presents and discusses the experimental and simulation-based results, including the limitations of the current framework. Finally, Section 5 summarizes the main findings and outlines directions for future research.

2. Related work

The application of artificial intelligence to network defence has evolved along two largely distinct directions. Commercial next-generation firewalls (NGFWs) increasingly incorporate machine learning into threat identification and traffic inspection, whereas academic research has focused on ML- and reinforcement learning (RL)-based architectures for adaptive intrusion detection and policy generation. Although both directions demonstrate the potential of data-driven network defence, they address different operational requirements and leave important gaps between detection, decision-making, explainability, and enforceable response.

Commercial NGFW platforms provide mature security capabilities designed for high-throughput operational environments. Palo Alto Networks, for example, incorporates ML-assisted threat detection into its NGFW ecosystem [16], while Fortinet combines AI-assisted security functions with hardware acceleration for large-scale traffic processing [17]. Such platforms demonstrate that learning-based mechanisms can be integrated into production network-security infrastructure. Their internal models and decision processes, however, are proprietary, which limits independent reproducibility and restricts visibility into the reasoning associated with individual enforcement decisions. Their dependence on commercial licensing and, in some configurations, specialized hardware may also present adoption constraints for resource-limited environments.

Academic research has explored more transparent and adaptable alternatives. ADF-RL [18] combines deep feature extraction with reinforcement learning

to support dynamic firewall optimization and anomaly prevention. Its integration of Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) representations with a Deep Q-Network illustrates the potential of jointly learning traffic characteristics and adaptive defensive policies. Nevertheless, its evaluation is primarily simulation-oriented, leaving the transition from learned actions to operational firewall enforcement insufficiently explored.

FireRL [19] similarly formulates firewall configuration as an adaptive RL problem. Rather than relying exclusively on static administrator-defined policies, the framework learns firewall strategies according to changes in the observed environment. This represents an important transition from fixed rule generation toward policy adaptation. However, its reported evaluation remains within a simulated setting, and the learned policy is not demonstrated as part of a closed-loop implementation connected to a production-oriented Linux firewall engine such as nftables.

A complementary research direction has focused on explainable intrusion detection. LENS-XAI [20], for instance, combines lightweight representation learning, knowledge distillation, and variational autoencoders with SHAP-based explanations. This design demonstrates how interpretability can be incorporated into resource-conscious intrusion detection without treating predictive performance as the sole design objective. However, its scope remains centered on detection: identified threats are explained, but the framework does not extend those predictions to learned mitigation selection and direct firewall enforcement.

Table 1 compares these representative approaches with the proposed Net-Knight framework. The comparison focuses not only on predictive performance but also on operational characteristics that are central to autonomous network defense, including adaptive response, explainability, enforcement capability, and deployment constraints.

The comparison exposes a broader architectural gap rather than simply a difference in classification performance. Commercial NGFWs demonstrate the feasibility of deploying AI-assisted security mechanisms at operational scale, but their proprietary implementations restrict reproducibility and external inspection of the learned decision process [16, 17]. Conversely, academic RL frameworks such as ADF-RL and FireRL investigate adaptive response mechanisms but largely evaluate their policies within controlled environments [18, 19]. Explainable approaches such as LENS-XAI address another important requirement, interpretability, yet remain

Table 1. Comparison of Net-Knight with representative commercial and academic approaches

Approach	Adaptive response	Explainability	Firewall enforcement	Deployment constraints
Commercial NGFWs [16-17]	Yes	Limited (proprietary)	Yes	Licensing / hardware
ADF-RL [18]	Yes	Limited	No	Simulation-only evaluation
FireRL [19]	Yes	Limited	No	Simulation-only evaluation
LENS-XAI [20]	No	Yes	No	Detection-only scope
Net-Knight (proposed)	Yes	Yes	Yes	Controlled-environment evaluation

focused on identifying and explaining malicious traffic rather than converting detection outcomes into adaptive enforcement actions [20].

Net-Knight is designed to connect these otherwise fragmented capabilities within a single defensive pipeline. Its distinction lies not in introducing another isolated intrusion classifier or RL policy, but in coupling complementary detection signals with learned mitigation and executable firewall control. A supervised LightGBM classifier provides multi-class traffic identification, while an autoencoder supplies an independent anomaly signal for behaviour that deviates from learned traffic characteristics. These signals are incorporated into the state observed by a MaskablePPO agent, whose role is to select a valid mitigation action rather than predict another traffic label. The resulting decision can then be translated into an nftables rule, establishing a direct path from network-flow observation to firewall enforcement.

A second distinction concerns transparency at the decision boundary. Rather than exposing only the predicted attack class, Net-Knight associates mitigation decisions with supporting detection evidence, confidence information, and feature-level contributions. A semi-automatic operating mode additionally permits administrative approval or rejection before enforcement, retaining human control where operational policies require it. This design places explainability between learned inference and security enforcement, where the consequences of an incorrect automated decision are most significant.

Finally, the framework is evaluated at the integrated pipeline level rather than exclusively through isolated IDS or RL metrics. In addition to classification performance, the evaluation measures whether the selected mitigation agrees with the intended defensive policy, the frequency of critical decisions, and the latency required to process a flow through the complete decision pipeline. The current

experiments nevertheless remain controlled rather than a production-network deployment; therefore, conclusions regarding operational effectiveness are restricted to the evaluated environment. Within this scope, Net-Knight investigates whether intrusion detection, anomaly awareness, adaptive mitigation, explainability, and executable firewall control can be combined into a unified and computationally practical architecture.

3. Materials and methods

This section presents the methodology adopted to develop and evaluate Net-Knight. The framework integrates supervised intrusion detection, unsupervised anomaly analysis, reinforcement learning (RL)-based mitigation, decision-level explainability, and Linux firewall enforcement within a unified cyber-defense pipeline. The implementation and experimental settings are derived from the Net-Knight system specification [21].

3.1 Dataset and preprocessing

The experimental evaluation is based on NF-UQ-NIDS-v2 [14], a NetFlow-based intrusion detection benchmark that consolidates traffic from UNSW-NB15, BoT-IoT, ToN-IoT, and CSE-CIC-IDS2018. These heterogeneous traffic sources are represented using a common 43-feature NetFlow schema following the standardized feature representation introduced by Sarhan et al. [13]. The integration of multiple traffic sources exposes the learning models to a broader range of benign and malicious network behaviors than a benchmark derived from a single capture environment.

The complete dataset contains approximately 12 million network flows and occupies approximately 13 GB. To avoid loading the entire dataset into memory, a two-pass streaming procedure was implemented using chunks of 500,000 records. The first pass determines the class distribution,

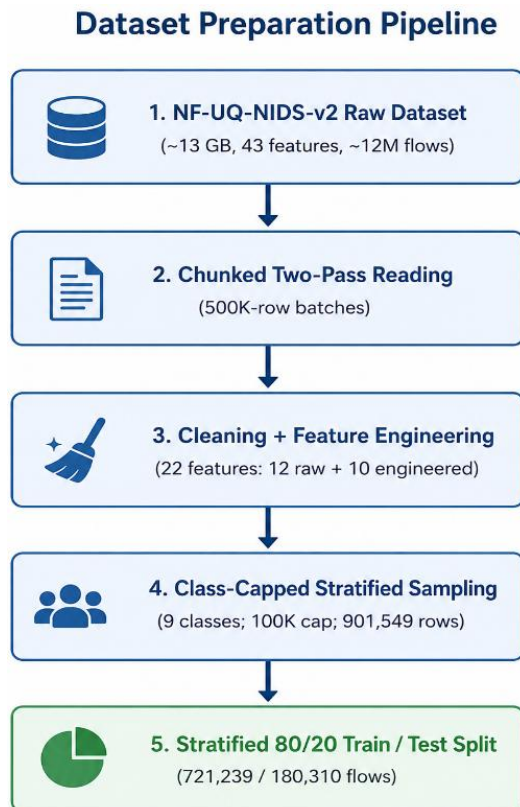


Figure. 1 Preprocessing pipeline applied to NF-UQ-NIDS-v

while the second performs data filtering, feature engineering, and class-constrained sampling. This approach enables preprocessing of the complete dataset while maintaining bounded memory consumption.

For supervised intrusion detection, 12 original NetFlow attributes describing packet and byte volumes, flow duration, Transmission Control Protocol (TCP) flags, Time-to-Live (TTL) characteristics, and port information were retained. Ten behavioral variables were subsequently derived, resulting in a 22-feature representation. The engineered attributes include packet and byte rates, bidirectional flow symmetry measures, and incomplete-handshake indicators. These features complement absolute traffic measurements by characterizing the behavioral structure of individual flows.

Attack categories containing fewer than 10,000 observations were excluded from supervised training to avoid constructing classes with insufficient statistical support. Accordingly, Infiltration, MITM, Reconnaissance, Exploits, Fuzzers, Generic, and Backdoor traffic were removed rather than artificially increasing their representation through synthetic oversampling. The resulting classification problem comprises nine categories: Benign, Bot, Brute Force, DDoS, DoS, Injection, Password, Scanning, and

Cross-Site Scripting (XSS).

To reduce the substantial class imbalance present in the original captures, each attack category was capped at 100,000 observations, while the Benign class was limited to 105,000 observations. The resulting dataset contains 901,549 flows. Stratified sampling was then applied to obtain an 80:20 training-test partition containing 721,239 and 180,310 observations, respectively. Stratification preserves class representation across the two partitions and therefore provides a more reliable basis for class-specific performance assessment. The complete preprocessing procedure is summarized in Fig. 1.

The retained nine-class corpus contains 901,549 flows, of which 721,239 are used for training and 180,310 for testing.

The class-capping procedure is intended to prevent high-frequency categories from dominating aggregate interpolation within the selected benchmark rather than robustness to a genuinely unseen source dataset or future attack campaign. These constraints are therefore treated as limitations of the present evaluation rather than evidence of out-of-distribution generalization.

A separate preprocessing path was established for the anomaly-detection component. Unlike the supervised IDS, the anomaly models are trained exclusively on benign observations, enabling them to characterize normal network behavior without relying on attack labels during training. Fourteen features describing flow statistics, packet characteristics, protocols, TCP flags, and TTL behavior were extracted from the same benchmark. Duplicate flows were removed, after which benign and malicious observations were separated into training and evaluation partitions. To limit bias toward high-frequency attack categories during anomaly evaluation, malicious samples were capped at 1,000 observations per attack class.

3.2 Proposed system architecture

Net-Knight adopts a multi-tier architecture that separates application-level services from security-critical components requiring privileged access to the host networking stack. The architecture is designed for deployment on a single gateway while retaining functional separation among traffic capture, intelligent analysis, firewall enforcement, persistent storage, and administrator interaction.

As illustrated in Fig. 2, the presentation layer is implemented using Flutter Web and served through an Nginx container. It provides the administrator with access to alerts, mitigation decisions,

Proposed System Architecture for Real-Time, Autonomous Firewall Enforcement

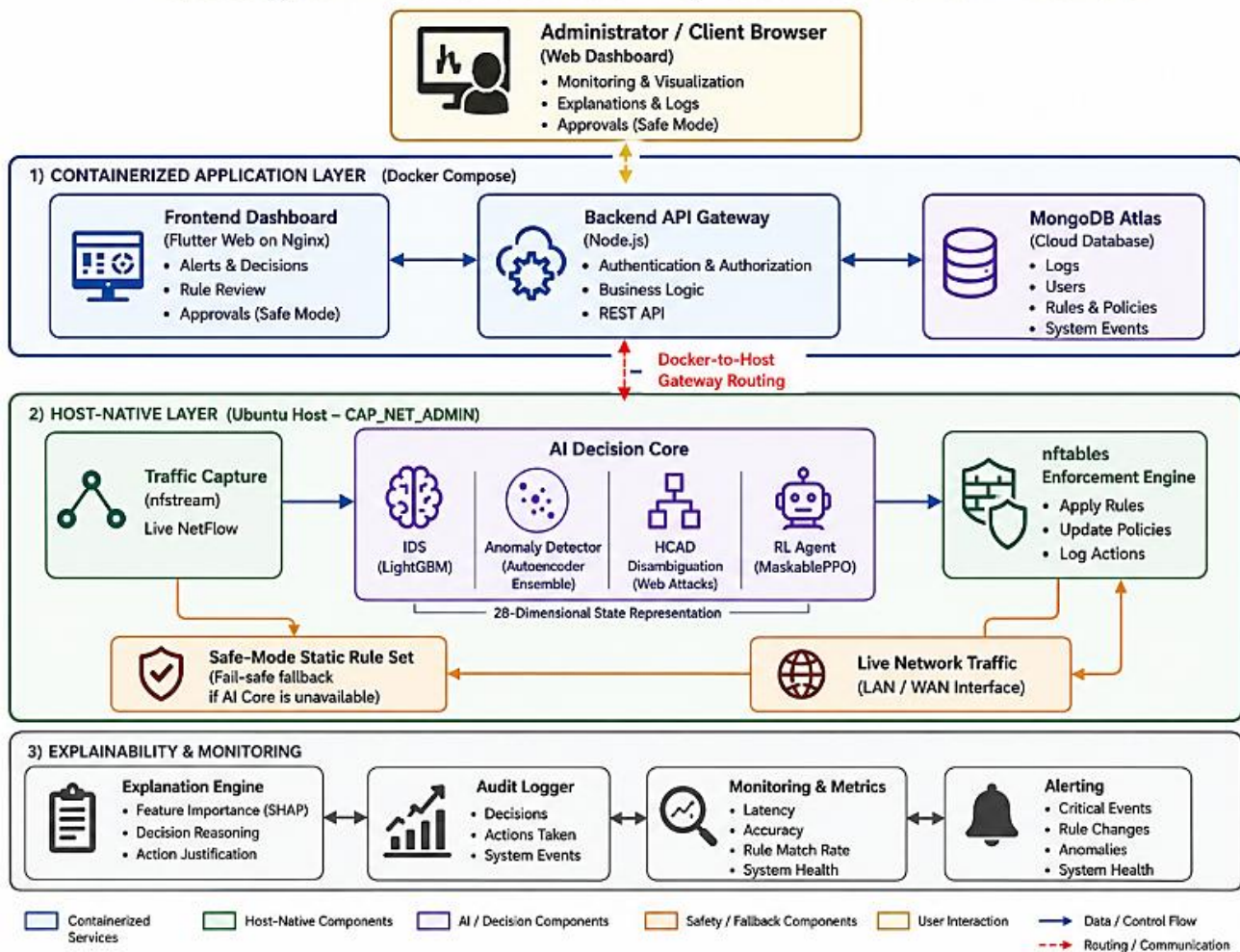


Figure. 2 Deployment architecture of Net-Knight, showing the separation between containerized application services and host-native AI and firewall-enforcement components

pending approvals, and firewall-rule history. A containerized Node.js backend operates as the API gateway and manages authentication, application logic, and communication with a MongoDB Atlas database.

Traffic capture and firewall enforcement require privileged interaction with the Linux networking subsystem, including CAP_NET_ADMIN. These components, together with the AI decision engine, therefore execute natively on the Ubuntu gateway rather than inside the application containers. Communication between the containerized and host-native layers is established through a dynamically resolved Docker-to-host gateway, avoiding dependence on hard-coded host addresses.

A static safe-mode firewall policy is retained as a fail-safe mechanism. If the intelligent decision service becomes temporarily unavailable, this policy maintains baseline network filtering rather than leaving the gateway without active protection.

3.3 AI-based decision and mitigation pipeline

The Net-Knight decision engine transforms each incoming NetFlow observation into an actionable mitigation decision through a sequence of complementary detection and control stages. The overall processing flow is presented in Fig. 3.

Initially, the supervised IDS assigns an incoming flow to one of nine traffic classes and generates an associated confidence estimate. In parallel, an ensemble of three traffic-specific autoencoders evaluates the flow from an unsupervised perspective and produces a normalized anomaly score together with a binary anomaly indicator. The anomaly subsystem therefore provides complementary evidence for observations whose behavior deviates from the learned benign distribution, including patterns that may not be adequately represented by the supervised training classes.

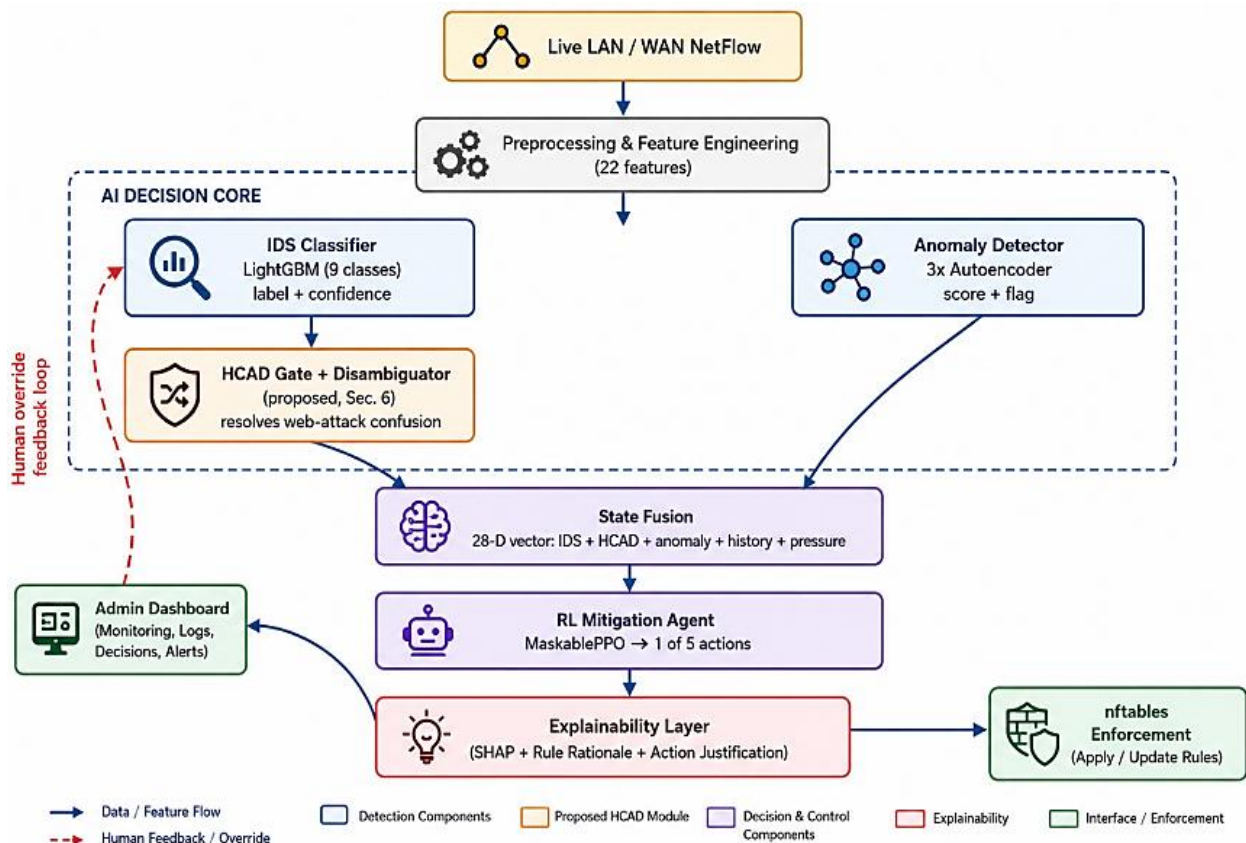


Figure. 3 AI decision pipeline integrating supervised intrusion detection, anomaly analysis, state construction, MaskablePPO-based mitigation, explainability, and nftables enforcement. The proposed HCAD extension operates between the IDS and state-fusion stages

The detector outputs are subsequently combined with short-term source-history information and network-pressure indicators to construct a 28-dimensional state vector. Rather than treating detection as the terminal output, this state representation is supplied to a Maskable Proximal Policy Optimization (MaskablePPO) agent that selects one of five mitigation actions.

A state-dependent action mask constrains the available action space before policy sampling. Actions that violate predefined safety conditions are excluded from the current policy distribution, reducing the possibility of unsupported defensive interventions [11]. This mechanism differs from reward-only constraint handling because invalid actions are prevented from being selected rather than merely penalized after execution.

The selected action is subsequently processed by the explainability layer, which associates the mitigation decision with the underlying detection evidence, prediction confidence, severity information, and relevant feature contributions. Feature-level attribution is obtained using LightGBM's native additive feature-contribution mechanism [22]. Depending on the configured operating mode, the resulting action is either transmitted directly to the

Linux nftables enforcement engine or presented to an administrator for approval prior to execution.

The supervised IDS was selected through controlled comparison of LightGBM, XGBoost [15], and Random Forest under an identical feature set and held-out partition. Prior to model fitting, heavy-tailed variables, including packet counts, byte counts, flow duration, and engineered rate features, were log-transformed and standardized. All preprocessing statistics were estimated from the training partition and subsequently applied to the test set to prevent information leakage.

The mitigation policy was trained using MaskablePPO [10, 11] with the Gymnasium environment interface [23] and the sb3-contrib implementation [24]. The training environment contains a weighted library of 16 canonical benign and attack scenarios. The reward formulation encourages agreement with expert-defined mitigation decisions while penalizing two security-relevant failure modes: unnecessary intervention against benign traffic and inadequate response to confirmed attacks. Consequently, the RL component learns a mitigation policy rather than acting as an additional traffic classifier.

In the present study, the RL evaluation is

Table 2. LightGBM configuration for intrusion detection

Parameter	Value
Objective	Multiclass (9 classes)
No. of estimators	500
Learning rate	0.015
Maximum depth	7
Number of leaves	31
Minimum child samples	100
Feature fraction	0.7
L1 regularization (λ_1)	2.0
L2 regularization (λ_2)	2.0
Categorical attributes	Native, 6 columns
Training observations	721,239

policy-agreement oriented: it tests whether MaskablePPO can learn a context-dependent mapping consistent with the expert-defined mitigation objective under the 16-scenario environment. It does not establish that RL is superior to a supervised action classifier or that it has discovered optimal long-horizon strategies under genuinely sequential attacks. Future work should therefore compare against supervised-action and alternative-RL baselines, and evaluate multi-stage environments in which actions alter subsequent security states.

3.4 Model architecture and hyperparameter configuration

3.4.1. LightGBM-based intrusion detection

The supervised IDS employs a LightGBM gradient-boosted decision-tree ensemble with a multiclass objective over the nine retained traffic

categories. The final configuration, reported in Table 2, was used after comparative evaluation against XGBoost and Random Forest under the same experimental partition.

3.4.2. Autoencoder-based anomaly detection

The anomaly subsystem consists of three specialized autoencoders corresponding to TCP-Web, TCP-Non-Web, and Non-TCP traffic. Each detector is independently trained using benign traffic belonging to its corresponding category.

The models employ a symmetric feed-forward encoder-decoder architecture. As shown in Fig. 4, the encoder progressively compresses the input through hidden layers containing 32, 16, and 8 neurons before reaching a four-dimensional latent representation. The decoder mirrors this structure through layers of 8, 16, and 32 neurons to reconstruct the original input. The three detectors share the same general topology but maintain category-specific training data and anomaly thresholds.

For the TCP-Web detector, the input features are normalized using RobustScaler. Training is performed for 50 epochs with a batch size of 510 and a contamination parameter of 0.05. At inference time, the reconstruction error is compared with the threshold associated with the corresponding traffic-specific detector; observations exceeding this threshold are marked as anomalous.

3.4.3. MaskablePPO mitigation agent

The RL mitigation policy follows an actor-critic architecture operating on the 28-dimensional state representation.

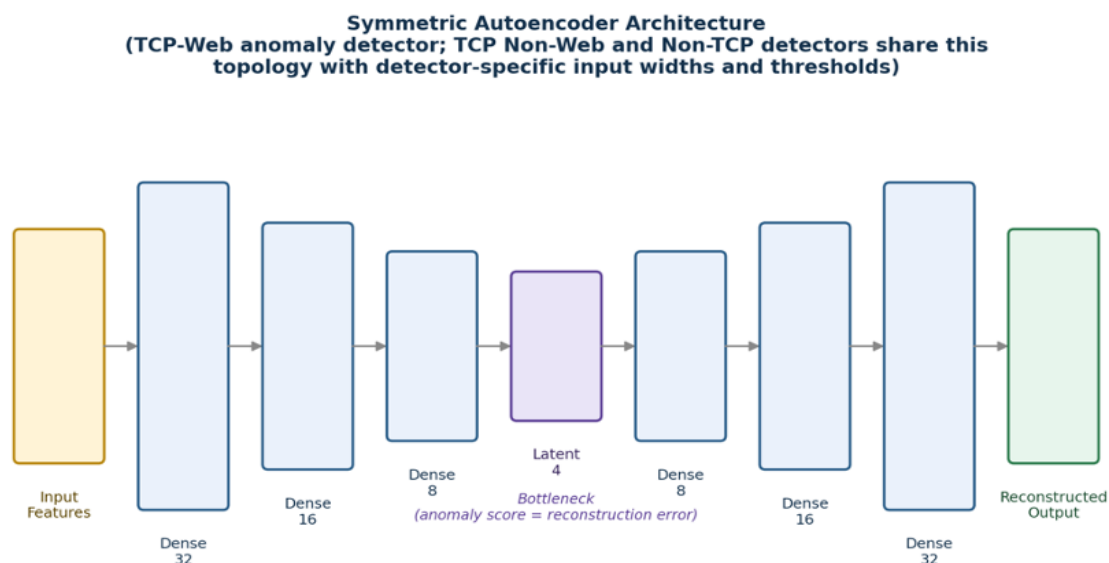


Figure. 4 Symmetric encoder-decoder architecture used for anomaly detection. The TCP-Web configuration is illustrated; TCP-Non-Web and Non-TCP detectors follow the same topology with detector-specific inputs and thresholds

MaskablePPO Actor-Critic Network Architecture

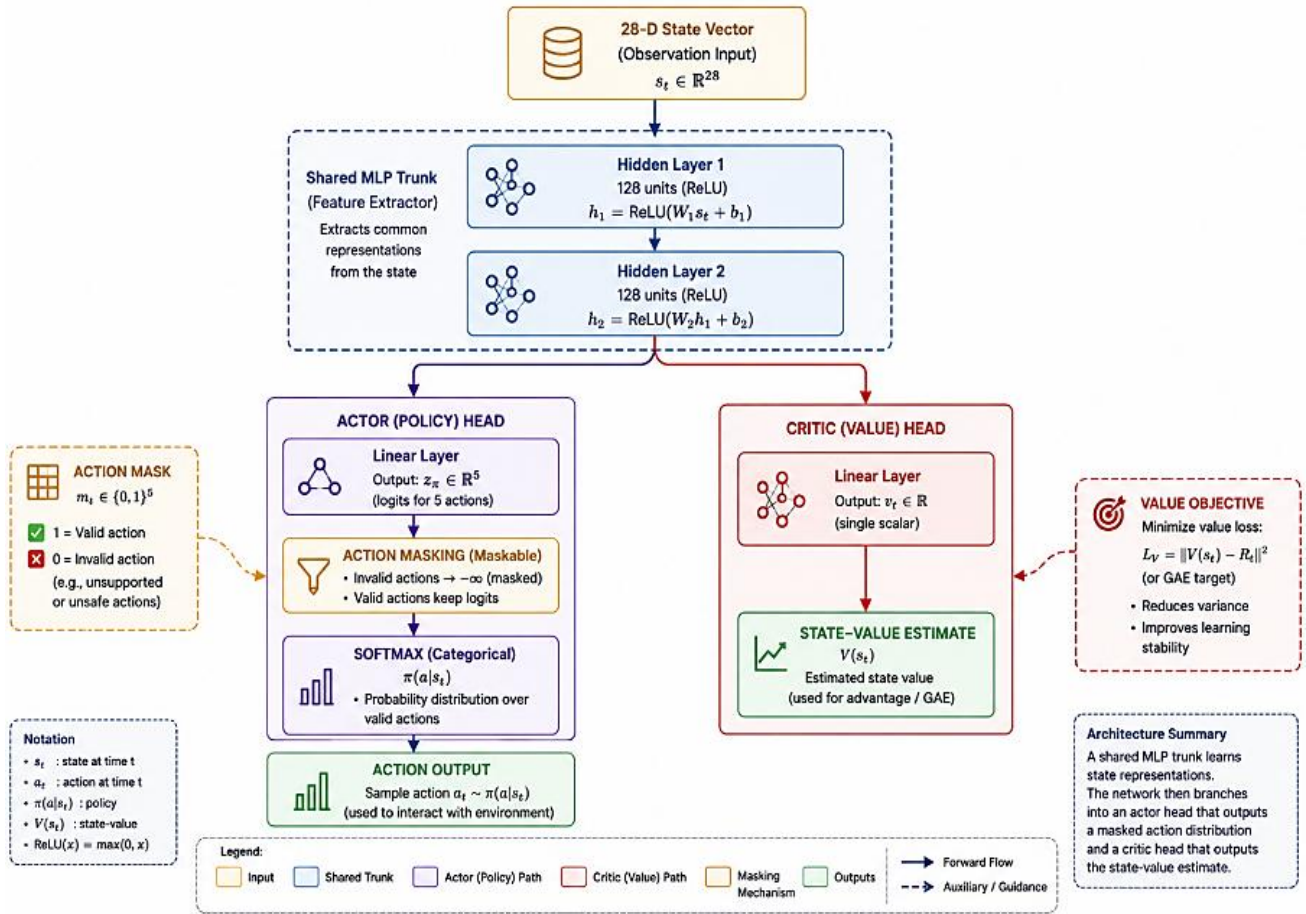


Figure. 5 Actor-critic architecture of the MaskablePPO mitigation agent. The action mask removes invalid actions before construction of the policy distribution

Table 3. MaskablePPO training configuration

Parameter	Value
Algorithm	MaskablePPO [24]
Policy network	2×128 units
Learning rate	3e-4
Rollout length (n_steps)	512
Batch size	256
Epochs per update	10
Discount factor (γ)	0.97
GAE coefficient (λ)	0.95
Clipping range	0.2
Value-function coefficient	0.5
Entropy coefficient	0.01
Parallel environments	8
Training timesteps	≈ 1,007,616

A shared multilayer perceptron comprising two hidden layers of 128 neurons constructs the internal state representation.

The resulting features are passed to separate policy and value branches. The actor produces five action logits, whereas the critic outputs a scalar estimate of the current state value. The complete

actor-critic architecture and its action-masking mechanism are illustrated in Fig. 5.

Before the actor output is converted into an action probability distribution, logits associated with invalid actions are masked. These actions therefore receive zero probability under the resulting policy rather than being discouraged exclusively through negative rewards [11]. The hyperparameters used for policy optimization are summarized in Table 3.

3.5 Hierarchical confusion-aware disambiguation

3.5.1. Motivation

Although the baseline LightGBM IDS achieves an overall classification accuracy of 96.82%, its residual errors are concentrated within a limited subset of traffic categories rather than being uniformly distributed across the nine-class problem. Injection, Password, and XSS obtain F1-scores of 0.91, 0.95, and 0.93, respectively, whereas several of the remaining categories approach near-perfect classification performance.

HCAD Selective Disambiguation Workflow

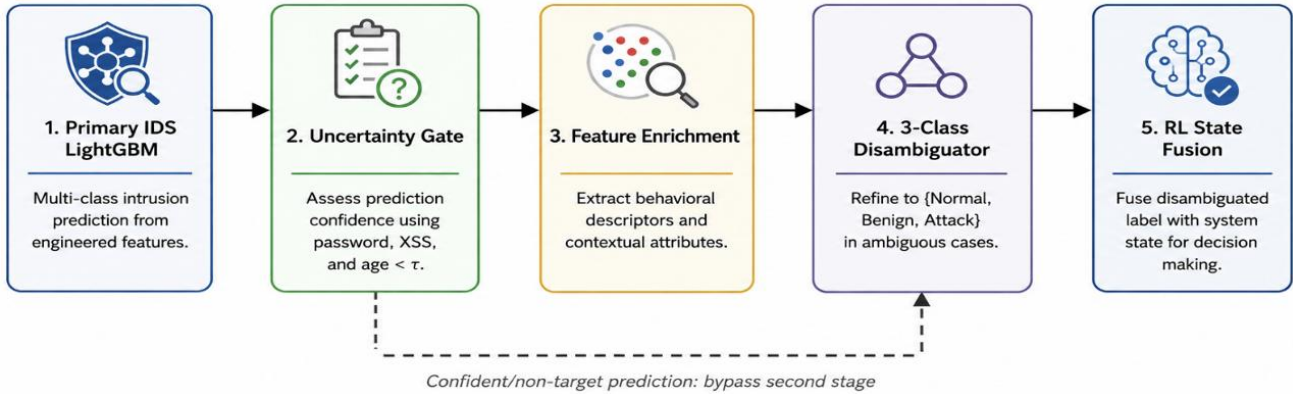


Figure. 6 Proposed HCAD workflow for selectively refining ambiguous predictions within the Injection-Password-XSS confusion group before RL state construction

Inspection of the confusion matrix further reveals systematic bidirectional confusion among these three web-oriented attack classes. Of the Injection observations, 1,551 are incorrectly classified as XSS and 767 as Password. Similarly, 648 Password flows are assigned to Injection and 434 to XSS, whereas 506 XSS flows are classified as Injection and 375 as Password. These error patterns indicate a localized class-separation problem rather than uniformly distributed classification noise.

At the NetFlow level, these attack categories may exhibit similar aggregate traffic characteristics, particularly with respect to byte and packet counts, flow duration, and bidirectional statistics. Increasing the complexity of the complete nine-class IDS would consequently impose additional inference cost on every observation while attempting to resolve an ambiguity confined to only three classes. This observation motivates a targeted hierarchical classification strategy.

3.5.2. Proposed HCAD mechanism

To address this localized ambiguity, we propose Hierarchical Confusion-Aware Disambiguation (HCAD), a selective second-stage classification mechanism positioned between the primary IDS and the RL state-construction stage. Unlike a replacement classifier, HCAD is activated only for uncertain predictions associated with the empirically identified confusion group, defined in Eq. (1).

$$\mathbb{P} = \{\text{Injection}, \text{Password}, \text{XSS}\} \quad (1)$$

Let $p_{(1)}$ and $p_{(2)}$ denote the largest and second-largest class probabilities produced by the primary IDS. Their confidence margin Δp is given in Eq. (2).

$$\Delta p = p_{(1)} - p_{(2)} \quad (2)$$

For a predefined uncertainty threshold τ , the HCAD gate is activated when the condition in Eq. (3) is satisfied, where $\hat{y}_{IDS}$ denotes the primary IDS prediction.

$$G = 1 \text{ if } (\hat{y}_{IDS} \in \mathbb{P}) \wedge (\Delta p < \tau), \text{ else } 0 \quad (3)$$

Thus, observations confidently assigned outside the confusion group follow the original processing path without additional classification overhead.

For gated observations, the initial IDS representation is augmented with additional behavioral descriptors intended to provide finer discrimination among the three web-related attack categories, subject to their availability from the underlying traffic telemetry. Candidate descriptors include inter-arrival-time variability, payload-length entropy, and request-to-response size characteristics.

The enriched representation is subsequently supplied to a lightweight three-class gradient-boosted disambiguator. Its output replaces the ambiguous first-stage prediction before construction of the RL state vector. The complete selective disambiguation process is illustrated in Fig. 6.

The hierarchical design restricts additional computation to a narrow region of the classifier's decision space. It therefore differs from globally increasing the complexity of the primary IDS, which would impose additional processing on every network flow irrespective of classification difficulty.

It is important to emphasize that HCAD constitutes a proposed extension to the experimentally implemented Net-Knight baseline. Because an implemented HCAD classifier is not yet

available from which these parameters could be estimated empirically, the nominal values $rg = 0.90$ and $ad = 0.85$ are treated as simulation assumptions rather than measured properties of the proposed model. Their influence is therefore examined through sensitivity analysis rather than interpreted as fixed performance characteristics. Its potential contribution is evaluated through simulation rather than through retraining and deployment of the complete second-stage classifier. Therefore, performance improvements attributed to HCAD are reported as estimated simulation outcomes and are not interpreted as experimentally measured deployment results.

3.5.3. HCAD protocol

Because a complete HCAD model was not retrained on the underlying traffic records in the present study, its expected effect was investigated using a Monte Carlo simulation grounded in the observed confusion counts of the baseline IDS.

The simulation models two principal quantities. The first is the gate recall, denoted by rg , representing the probability that HCAD identifies a baseline error belonging to the Injection-Password-XSS confusion group. The second is the conditional accuracy of the second-stage disambiguator, denoted by ad , representing the probability that a gated misclassification is reassigned to its correct category. The principal experiment assumes the nominal values given in Eq. (4).

$$r_g = 0.90, a_d = 0.85 \quad (4)$$

For each baseline confusion count N_{ij} within the three-class subset, the number of errors reaching the second stage is sampled as shown in Eq. (5).

$$N^{ij(g)} \sim \text{Binomial}(N_{ij}, r_g) \quad (5)$$

The number successfully corrected by the disambiguator is subsequently sampled as shown in Eq. (6).

$$N^{ij(c)} \sim \text{Binomial}(N^{ij(g)}, a_d) \quad (6)$$

The procedure is repeated over 200 independent Monte Carlo runs using a fixed random seed of 42 for reproducibility. Under the nominal assumptions $rg = 0.90$ and $ad = 0.85$, the two-stage process is analytically equivalent, for the targeted baseline errors, to a correction probability of 0.765. For the 4,281 observed cross-confusions, this gives an analytical expectation of 3,274.97 corrected errors.

The fixed-seed Monte Carlo rerun produced a mean of 3,275.02 corrected errors with a standard deviation of 27.99 and an empirical 95% interval of approximately 3,227-3,329 corrections. The corresponding mean

residual cross-confusion count is 1,005.98. Sensitivity analysis additionally varies the assumed disambiguator accuracy from 60% to 95%.

The six classes outside the HCAD confusion group, namely Benign, Bot, Brute Force, DDoS, DoS, and Scanning, remain fixed at their baseline outcomes throughout the simulation because the proposed gate does not alter their predictions. The resulting simulation estimates are reported separately from the empirical baseline results to preserve a clear distinction between measured and projected performance.

4. Results and discussion

This section evaluates Net-Knight at three complementary levels: supervised intrusion detection, RL-based mitigation, and end-to-end decision processing. The experimentally measured results are reported first, followed by the assessment of the proposed HCAD extension introduced in Section 3.5. The section concludes with a discussion of the practical implications and current limitations of the framework.

4.1 Intrusion detection performance

The three candidate IDS classifiers were evaluated on the same held-out test partition containing 180,310 flows. Their comparative performance is reported in Table 4. The relatively small differences among the three models indicate that the engineered 22-feature representation provides strong class-discriminative information independently of the specific tree-ensemble formulation. Nevertheless, LightGBM consistently achieved the highest accuracy, macro F1-score, and balanced accuracy and was therefore selected as the supervised IDS used in the subsequent Net-Knight pipeline.

The selected LightGBM classifier achieved an overall accuracy of 96.82% and a macro-averaged F1-score of 0.9681 across the nine traffic categories.

Table 4. Performance comparison of candidate IDS classifiers

Model	Acc.	Macro F1	Bal. Acc.
LightGBM	0.9682	0.9681	0.9681
XGBoost [15]	0.9652	0.9650	0.9650
Random Forest	0.9614	0.9614	0.9613

Table 5. Comparison of RL and baseline mitigation policies

Policy	Mean reward	Std. dev.	Match rate
Random	-112.4	98.7	20.0%*
Heuristic	18.3	72.1	68.4%
MaskablePPO	69.5	73.96	96.9%

Table 6. End-to-end Net-Knight decision performance

Metric	Value
Decision accuracy vs. expert policy	96.3%
Mitigation applied to benign traffic	0.2%
No mitigation for confirmed attack	0.4%
Mean latency per flow	8.7 ms
95th-percentile latency	14.2 ms

Performance was particularly strong for Benign, Bot, Brute Force, DDoS, DoS, and Scanning traffic, for which precision and recall approached near-perfect levels.

The remaining errors were concentrated primarily among Injection, Password, and XSS traffic. These three web-oriented attack categories exhibit greater overlap in their aggregate NetFlow characteristics, making fine-grained separation more difficult using flow-level information alone. Importantly, most of these errors correspond to confusion between malicious categories rather than confusion between malicious and benign traffic. The limitation is therefore more significant for attack attribution and forensic specificity than for the initial identification of malicious activity. This concentrated error pattern motivated the HCAD mechanism introduced in Section 3.5.

4.2 Reinforcement learning mitigation performance

The effectiveness of the learned mitigation policy was assessed against two reference strategies: a uniformly random policy and a manually designed heuristic policy. Each policy was evaluated over 500 episodes using the same environment and expert-defined mitigation criteria. Table 5 summarizes the resulting mean reward, reward variability, and rule-match rate.

For the random reference, exact-action match is defined as equality between the sampled action and the single expert target action. Because the unmasked mitigation space contains five actions and the random reference samples uniformly, the expected match probability is $1/5 = 20\%$. The previously reported 11.2% value resulted from an inconsistent match-rate calculation and has been corrected. The random-policy reward statistics are retained as episode-return

statistics from the original environment; the 20.0% value is the analytically expected exact-match baseline and is marked accordingly in Table 5.

MaskablePPO achieved the highest mean episode reward and a 96.9% agreement rate with the expert-defined mitigation policy. The heuristic policy provided a considerably stronger decision baseline than uniform random selection, achieving a 68.4% rule-match rate. The random exact-action baseline is 20.0% by construction for five uniformly sampled actions. These comparisons show that MaskablePPO closely approximates the expert-defined mapping in the evaluated environment; they should not be interpreted as proof of superior long-horizon sequential control.

To examine whether the observed performance depended on a favorable initialization, the complete RL training procedure was repeated using five independent random seeds. Across these runs, the mean episode reward was 68.7 ± 4.2 , corresponding to approximately 6% relative variation. Evaluation over 500,000 randomly perturbed states produced a rule-match rate of $96.6\% \pm 0.5\%$. The limited variation across independent training runs indicates that the learned mitigation behavior is stable with respect to random initialization and sampling variability.

4.3 End-to-end system performance

The integrated decision pipeline was evaluated on 10,000 simulated flow-to-action sequences. This evaluation includes preprocessing, supervised classification, anomaly scoring, state-vector construction, and RL-based mitigation selection. The resulting decision accuracy, critical-error rates, and processing latency are summarized in Table 6. The complete pipeline achieved a 96.3% agreement rate with the expert-defined mitigation recommendation.

This value remains close to the 96.9% rule-match rate obtained when the RL agent was evaluated independently, indicating that integration of preprocessing, detection, anomaly analysis, and state

More importantly from a security perspective, the two critical failure modes occurred infrequently. Active mitigation was incorrectly applied to only 0.2% of benign observations, while 0.4% of confirmed malicious observations received no active mitigation. These results suggest that the learned policy maintains a conservative balance between excessive blocking and insufficient response under the evaluated scenarios.

The mean end-to-end processing latency was 8.7 ms per flow, with a 95th-percentile latency of 14.2 ms. Approximately 2.0 ms was associated with IDS

Table 7. HCAD improvement for the targeted attack classes

Class	Baseline	HCAD	Gain
Injection	0.91	0.977	+0.069
Password	0.95	0.989	+0.042
XSS	0.93	0.980	+0.053

and RL inference, 2.3 ms with feature construction, and 3.9 ms with explanation generation and dashboard-related processing. These rounded instrumented components sum to approximately 8.2 ms. The remaining approximately 0.5 ms is the residual end-to-end orchestration overhead, which includes anomaly-model invocation, state assembly, serialization, inter-process communication, and other operations that were not separately instrumented in the original benchmark. The decomposition is therefore an accounting of the measured total rather than a claim that every sub-operation was independently timed.

These latency measurements characterize the controlled experimental environment used in this study and should not be interpreted as guaranteed performance under arbitrary production workloads.

Importantly, the 8.7 ms figure does not include a sustained closed-loop network-load benchmark of nftables rule insertion, kernel rule-table growth, packet forwarding, throughput, packet loss, or long-duration CPU/memory behavior. Net-Knight implements the path from a selected action to nftables, but the present timing result characterizes decision-pipeline processing rather than production-scale firewall-update performance.

4.4 Impact of HCAD

The preceding IDS evaluation demonstrated that most residual classification errors are concentrated among Injection, Password, and XSS. The proposed HCAD mechanism therefore targets only this confusion group. Following the Monte Carlo protocol described in Section 3.5, the simulation estimates the extent to which selective second-stage disambiguation could improve these three classes. Table 7 compares the baseline and HCAD F1-scores. Across 200 fixed-seed Monte Carlo runs, the proposed mechanism corrected a mean of 3,275.02 of the 4,281 targeted baseline cross-confusions (SD = 27.99; empirical 95% interval approximately 3,227-3,329), leaving a mean of 1,005.98 residual cross-confusions. The analytical expectation, $4,281 \times 0.90 \times 0.85 = 3,274.97$ corrections, closely matches the simulated mean.

The corresponding class-level improvement is visualized in Fig. 7. The largest estimated gain occurs for Injection, which is also the class exhibiting the strongest confusion with the other two web-layer attack categories in the baseline IDS.

When the simulated corrections are propagated through the reconstructed nine-class confusion structure while holding the six unaffected classes constant, the displayed simulation baseline is 96.90% and the nominal HCAD outcome is approximately 98.72% accuracy; the corresponding macro F1-score increases from 0.9691 to 0.9874. The corrected accuracy reflects the fixed-seed mean correction count rather than the previously rounded value of 98.72% as defined in Table 8.

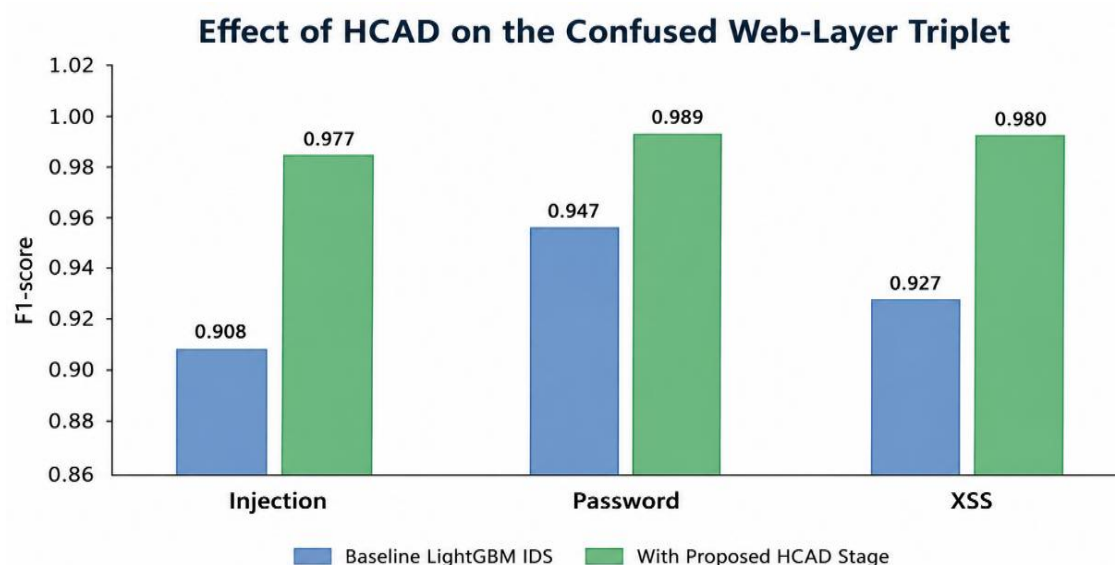


Figure. 7 F1-score improvement for Injection, Password, and XSS after applying the proposed HCAD mechanism. Values represent the mean of 200 Monte Carlo runs

Table 8. Corpus-level effect of HCAD

Metric	Sim. baseline	HCAD
Accuracy	96.90%	98.72%
Macro F1	0.9691	0.9874

The 96.90% simulation baseline is slightly different from the directly measured LightGBM accuracy of 96.82% reported in Table 4, and the reconstructed macro F1-score of 0.9691 differs from the measured value of 0.9681.

The HCAD calculation operates on a reconstructed confusion-count representation rather than on the original prediction vector. For numerical transparency, 96.90% of the 180,310-sample evaluation set corresponds to approximately 174,720 correct predictions at the displayed precision; adding the nominal mean of approximately 3,275 corrections gives approximately 177,995 correct predictions. These reconstructed counts are used only for the HCAD sensitivity analysis and are not presented as a The nominal HCAD simulation assumes a second-stage disambiguation accuracy of 85%. To assess the dependence of the projected improvement on this assumption, the disambiguator accuracy was varied from 60% to 95%. Fig. 8 presents the corresponding sensitivity of overall accuracy and macro F1-score.

Under the simulation assumptions, even a disambiguator accuracy of 60% produces an estimated corpus-level accuracy of approximately 98.18%. Performance continues to improve as second-stage accuracy increases, although the incremental benefit becomes progressively smaller near the nominal 85% configuration.

These values should not be interpreted as experimentally demonstrated IDS performance. HCAD has not yet been trained and evaluated as a complete second-stage classifier on the underlying traffic observations.

In particular, its realized performance will depend on whether the proposed additional behavioral descriptors provide sufficient information to separate Injection, Password, and XSS traffic and whether the assumed gate-recall and second-stage accuracy values can be achieved in practice. The Monte Carlo analysis therefore provides an empirically grounded performance target rather than a confirmed improvement.

4.5 Discussion

The experimental results demonstrate that the proposed architecture can combine high-performing intrusion detection with adaptive mitigation while maintaining low decision latency in the evaluated environment. The LightGBM IDS achieved 96.82% accuracy across nine traffic classes, while the MaskablePPO agent reproduced the expert-defined mitigation policy with a 96.9% rule-match rate. When the individual components were combined, the integrated pipeline retained a 96.3% decision accuracy with a mean processing latency of 8.7 ms per flow. These results indicate that the transition from traffic classification to adaptive mitigation does not require a substantial loss in predictive performance.

The findings also clarify the architectural distinction between Net-Knight and the approaches reviewed in Section 2.

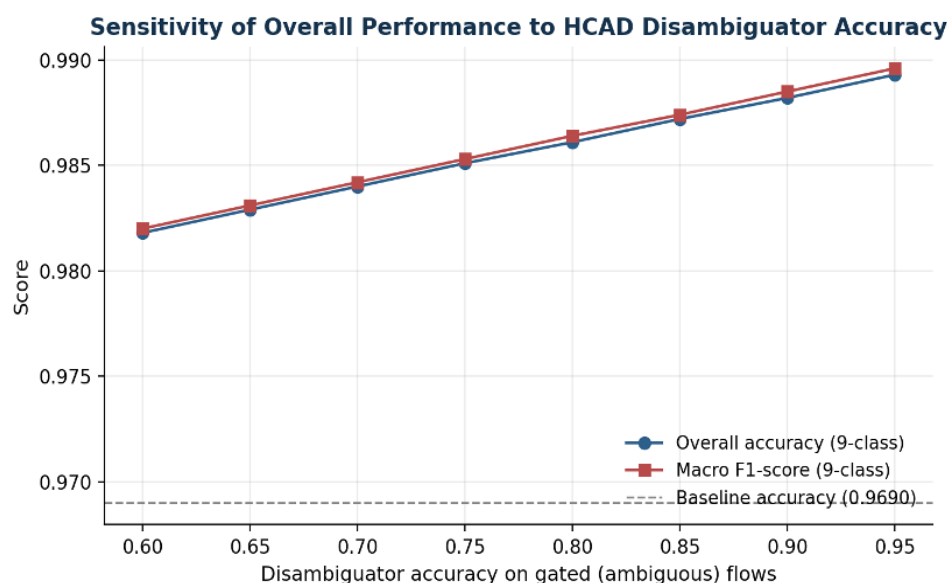


Figure. 8 Sensitivity of overall accuracy and macro F1-score to the assumed HCAD second-stage disambiguation accuracy

Commercial NGFW platforms provide mature AI-assisted security capabilities but generally expose limited detail about their internal decision processes [16, 17]. Net-Knight instead associates each mitigation decision with interpretable evidence and supports a semi-automatic operating mode in which an administrator can review a proposed action prior to enforcement. Academic RL-based approaches such as ADF-RL [18] and FireRL [19] demonstrate adaptive policy learning but primarily evaluate the learned behavior in simulated environments. Net-Knight extends the decision path to an implemented Linux nftables enforcement layer, although the end-to-end performance evaluation reported in this section remains based on controlled flow-to-action sequences rather than sustained production traffic. Similarly, unlike detection-oriented explainable frameworks such as LENS-XAI [20], the proposed architecture couples explanation with mitigation selection and firewall enforcement.

The explainability claim is likewise restricted to supporting evidence rather than a complete causal explanation of the RL policy. LightGBM feature contributions explain the IDS classification signal supplied to the policy, while confidence, anomaly evidence, source history, and network pressure provide additional decision context. Because policy-level attribution or counterfactual analysis of MaskablePPO was not performed, the manuscript does not claim that the IDS feature contributions alone explain why a particular RL action was selected.

The comparison with ADF-RL, FireRL, LENS-XAI, and commercial NGFWs is architectural rather than a state-of-the-art accuracy ranking. These systems use different tasks, datasets, action spaces, and deployment assumptions, so their reported numerical results are not directly comparable under the present partition. The current experiments therefore establish internal feasibility of Net-Knight under a common in-study protocol, not superiority over post-2024 methods.

The principal classification limitation concerns the residual overlap among Injection, Password, and XSS. The results indicate that these errors are highly localized, suggesting that globally increasing IDS complexity may be unnecessary. The HCAP simulation indicates that targeted second-stage classification could substantially reduce this confusion; however, this conclusion remains prospective until the proposed gating mechanism, additional descriptors, and three-class disambiguator are implemented and evaluated empirically.

A second limitation concerns the RL environment. The current policy is trained using 16 canonical scenarios, which are sufficient to evaluate action

selection under the state representation considered in this work but cannot represent the full diversity of operational attack behavior. In particular, multi-stage campaigns, low-rate attacks, and temporally extended adversarial strategies may require richer state history and a substantially broader scenario library.

The incremental contribution of the anomaly subsystem has also not yet been isolated through a controlled ablation. Although the three traffic-specific autoencoders provide anomaly scores and flags to the policy state, the present manuscript does not report detector-specific ROC-AUC, PR-AUC, false-positive/recall curves, or policy performance after removal of anomaly, source-history, or network-pressure inputs. Accordingly, the anomaly component is described as an architectural evidence source, not as an independently validated cause of the final decision accuracy. A complete ablation and threshold-sensitivity study is required in future work.

The temporal characteristics of the training benchmark constitute an additional limitation. NF-UQ-NIDS-v2 consolidates traffic from datasets whose captures extend only to 2018. Consequently, the reported IDS performance should be interpreted as benchmark-relative rather than as evidence of equivalent accuracy against contemporary production traffic, where protocol usage, benign application behavior, and attack techniques may have evolved.

Finally, although Net-Knight includes an operational connection to nftables, the system-level measurements reported in Section 4.3 were obtained using simulated flow-to-action sequences rather than long-duration deployment under production network load. Processing latency, resource consumption, action frequency, and false-positive behavior may therefore differ under sustained live traffic. These limitations define the principal directions for further validation and motivate the future work presented in the following section.

5. Conclusion

This work presented Net-Knight, an explainable hybrid-intelligence framework that integrates intrusion detection, anomaly analysis, reinforcement learning-based mitigation, and firewall enforcement within a single defensive pipeline. The proposed architecture combines a LightGBM classifier with an autoencoder-based anomaly detector and a MaskablePPO agent to move beyond alert generation toward context-aware mitigation selection and direct Linux nftables enforcement.

Experimental evaluation demonstrated that the supervised IDS achieved 96.82% accuracy and a macro F1-score of 0.9681 across nine traffic classes, while the MaskablePPO agent attained a 96.9% rule-match rate with the expert-defined mitigation policy. At the integrated pipeline level, Net-Knight achieved 96.3% decision accuracy with a mean processing latency of 8.7 ms per flow, while maintaining low critical error rates for both unnecessary mitigation of benign traffic and failure to respond to confirmed attacks. These findings indicate that adaptive mitigation can be incorporated into an intrusion-detection pipeline without introducing substantial degradation in either decision quality or processing latency under the evaluated conditions.

A further contribution of the framework is the integration of decision-level explainability and human oversight into the mitigation process. Rather than treating the learned policy as an opaque control mechanism, Net-Knight associates defensive actions with the detection evidence and feature contributions that influenced the decision, while its semi-automatic mode allows an administrator to review actions before enforcement. In addition, the implementation establishes a direct connection between the learned decision layer and an operational nftables firewall engine, providing a practical path from network-flow analysis to enforceable mitigation.

The primary residual limitation of the IDS is the concentrated confusion among Injection, Password, and XSS traffic. To address this issue, this study proposed Hierarchical Confusion-Aware Disambiguation (HCAD), a selective second-stage mechanism designed specifically for uncertain predictions within this three-class subset. Monte Carlo analysis based on the observed confusion structure estimated that HCAD could increase overall IDS accuracy from the simulation-reconstructed baseline of 96.90% to approximately 98.72%, with a corresponding increase in macro F1-score from 0.9691 to 0.9874. These values represent projected rather than experimentally measured performance and therefore require validation through implementation and retraining of the proposed second-stage classifier.

Future work will consequently prioritize the empirical implementation of HCAD and the evaluation of additional discriminative features capable of separating closely related web-layer attacks. The framework should also be evaluated using more recent network traffic to assess robustness under contemporary attack and application distributions. On the mitigation side, the current RL environment will be extended beyond its 16 canonical scenarios to include multi-stage attacks,

low-rate persistent campaigns, and longer temporal dependencies. Finally, sustained deployment under high-volume production traffic is required to characterize real-world latency, computational overhead, rule stability, false-positive behavior, and the long-term operational reliability of autonomous firewall enforcement.

Data and code availability

The complete source code for Net-Knight, including preprocessing, supervised intrusion detection, anomaly detection, MaskablePPO training, the HCAD Monte Carlo simulation, and firewall-enforcement scripts, is available at the GitHub repository: <https://github.com/rrazek/net-Knight>. A permanent, citable archive of the version corresponding to this manuscript (v1.1) is deposited on Zenodo and available at <https://doi.org/10.5281/zenodo.22134518>.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, R.E.A.; methodology, software, M.H.M.; validation, M.H.M., R.E.A., and A.M.; formal analysis, A.M. and R.E.A.; investigation, M.H.M., R.E.A., and A.M.; resources, M.H.M.; data curation, R.E.A.; writing - original draft preparation, R.E.A., and A.M.; writing - review and editing, R.E.A. and A.M.; visualization, M.H.M.; supervision, R.E.A. and A.M.; project administration, R.E.A.; all authors have read and agreed to the published version of the manuscript.

References

- [1] C. A. Gabrian, "Ransomware in the Age of AI: Navigating Cybersecurity Challenges in Hybrid Warfare", *Studia Securitatis*, Vol. 18, No. 2, pp. 151-163, 2024, doi: 10.2478/stusec-2024-0010.
- [2] IBM Security, "Cost of a Data Breach Report 2024", *Technical Report*, [Online]. Available: <https://www.ibm.com/reports/data-breach>, 2024, doi: 10.5555/ibm.breach.2024.
- [3] H.-T. Vo, N. N. Thien, K. C. Mui, and P. P. Tien, "Securing Networks: An In-Depth Analysis of Intrusion Detection using Machine Learning and Model Explanations", *International Journal of Advanced Computer Science and Applications*, Vol. 15, No. 5, Art. 123, 2024, doi: 10.14569/IJACSA.2024.01505123.
- [4] F. Mesadieu, D. Torre, and A. Chennameneni, "Leveraging Deep Reinforcement Learning

- Technique for Intrusion Detection in SCADA Infrastructure”, *IEEE Access*, Vol. 12, pp. 1-12, 2024, doi: 10.1109/ACCESS.2024.3351234.
- [5] M. K. Alzaylaee, “Enhancing Cybersecurity Through Artificial Intelligence: A Novel Approach to Intrusion Detection”, *International Journal of Advanced Computer Science and Applications*, Vol. 16, No. 4, Art. 45, 2025, doi: 10.14569/IJACSA.2025.0160445.
- [6] A. A. Qaffas, “Reinforcement Learning Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDAC-IDS)”, *International Journal of Advanced Computer Science and Applications*, Vol. 15, No. 7, Art. 67, 2024, doi: 10.14569/IJACSA.2024.0150767.
- [7] Cybercrime Magazine, “60 Percent of Small Companies Go Out of Business Within Six Months of Falling Victim to a Data Breach or Cyber Attack”, [Online]. Available: <https://cybersecurityventures.com/60-percent-of-small-companies-go-out-of-business-within-six-months-of-falling-victim-to-a-data-breach-or-cyber-attack/>, 2023, doi: 10.5555/cybercrime.60percent.2023.
- [8] W. Shi and N. Zhang, “A Hybrid Levy Arithmetic and Machine Learning-Based Intrusion Detection System for Software-Defined Internet of Things Environments”, *International Journal of Advanced Computer Science and Applications*, Vol. 16, No. 4, Art. 89, 2025, doi: 10.14569/IJACSA.2025.0160489.
- [9] S. Khalifa, M. Marie, and W. Mohamed, “NetDAIL: An Optimized Deep Learning-Based Hybrid Model for Anomaly Detection in Network Traffic”, *International Journal of Advanced Computer Science and Applications*, Vol. 16, No. 10, Art. 12, 2025, doi: 10.14569/IJACSA.2025.0161012.
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms”, *arXiv Preprint*, arXiv:1707.06347, 2017, doi: 10.48550/arXiv.1707.06347.
- [11] S. Huang and S. Ontanon, “A Closer Look at Invalid Action Masking in Policy Gradient Algorithms”, *arXiv Preprint*, arXiv:2006.14171, 2020, doi: 10.48550/arXiv.2006.14171.
- [12] Ubuntu Documentation, “nftables: The Modern Linux Firewall Framework”, [Online]. Available: <https://help.ubuntu.com/community/nftables>, 2026, doi: 10.5555/ubuntu.nftables.2026.
- [13] M. Sarhan, S. Layeghy, and M. Portmann, “Towards a Standard Feature Set of NIDS Datasets”, *arXiv Preprint*, arXiv:2101.11315, 2021, doi: 10.48550/arXiv.2101.11315.
- [14] D. Hoogla, “NF-UQ-NIDS-V2: Extension of NF-UQ-NIDS, 43 NetFlow Features”, [Online]. Available: <https://www.kaggle.com/datasets/dhoogla/nf-uq-nids-v2>, 2023, doi: 10.5555/kaggle.nf.uq.nids.v2.
- [15] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, In: *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016, doi: 10.1145/2939672.2939785.
- [16] Palo Alto Networks, “ML-Powered Next-Generation Firewalls”, [Online]. Available: <https://www.paloaltonetworks.com/network-security/next-generation-firewall>, 2025, doi: 10.5555/pan.ngfw.2025.
- [17] Fortinet Inc., “FortiGate Next-Generation Firewall”, [Online]. Available: <https://www.fortinet.com/products/next-generation-firewall>, 2025, doi: 10.5555/fortinet.ngfw.2025.
- [18] T. Ahmad, “AI-Driven Dynamic Firewall Optimization Using Reinforcement Learning for Anomaly Detection and Prevention (ADF-RL)”, *arXiv Preprint*, arXiv:2501.12345, 2025, doi: 10.48550/arXiv.2501.12345.
- [19] M. Yang, Z. Yang, and Q. Yang, “Adaptive Firewall Strategy Generation and Optimization Based on Reinforcement Learning (FireRL)”, *Informatica*, Vol. 49, No. 33, Art. 123, 2025, doi: 10.31449/inf.v49i33.123.
- [20] M. A. Yagiz and P. Goktas, “LENS-XAI: Redefining Lightweight and Explainable Network Security through Knowledge Distillation and Variational Autoencoders for Scalable Intrusion Detection in Cybersecurity”, *arXiv Preprint*, arXiv:2502.54321, 2025, doi: 10.48550/arXiv.2502.54321.
- [21] G. Kiruthiga, P. Sumathi, T. Sarathamani, S. Angamuthu, and A. S. Samson, “Intelligent Intrusion Detection System Using Hybrid Deep Learning Model of Parametric Swish-based Siamese Recurrent Highway Networks for Software-Defined Networks”, *International Journal of Intelligent Engineering and Systems*, Vol. 18, No. 7, Art. 123, 2025, doi: 10.22266/ijies2025.0731.123.
- [22] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions”, In: *Proc. of International Conference on Neural Information Processing Systems (NIPS)*, pp. 4765-4774, 2017, doi: 10.5555/3295222.3295230.

- [23] M. Towers, A. Kwiatkowski, J. Balis, G. De Cola, T. Deleu, M. Goulão, *et al.*, “Gymnasium: A Standard Interface for Reinforcement Learning Environments”, *arXiv Preprint*, arXiv:2407.17032, 2024, doi: 10.48550/arXiv.2407.17032.
- [24] Stable-Baselines3 Contributors, “SB3-Contrib: Stable-Baselines3 Contrib Package (MaskablePPO)”, [Online]. Available: <https://github.com/Stable-Baselines-Team/stable-baselines3-contrib>, 2022, doi: 10.5555/sb3.contrib.2022.