# A Reinforcement Learning Framework for Adaptive Privacy–Security Trade-off in Proxy Networks

**Dr. N. Nithiyanandam[1], Dr. B. Rajalingam[2], Dr. Govinda Rajulu[3]. Dr. E.Sophiya[4], Ch. Aruna Reddy[5], Dr. R.Santhoshkumar[6]**

[3]Professor, [2,4,5]Associate Professor, [1,5]Assistant Professor

[1]Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, 603203

[2,4]Department of CSE, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India, 500100

[3]Department of Computer Science & Design, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India, 500100

[6]Department of CSE, Sree Dattha Group of Institutions, sheriguda, Ibrahimpatnam, Hyderabad, 501510

[1]nnithi81@gmail.com, [2]rajalingam35@gmail.com, [3]rajulug7@gmail.com, [4]venus.sophiya@gmail.com, [5]anucse0308@gmail.com, [6]santhoshkumar.aucse@gmail.com

**Abstract**

Proxy-based networks continue to serve as an essential part of anonymous communication, traffic routing, and security policy enforcement. However, with predefined security and privacy configurations, these networks find it difficult to respond to changing scenarios, emerging cyber threats, and user requirements. The paper proposes an adaptive solution to these issues by using reinforcement learning for proxy-based security and privacy optimization. In this scenario, proxy action choices are modeled as Markov decision processes. The RL agent selects the optimal strategies for privacy and security, such as encryption levels, traffic privacy, access mechanisms, and anomaly levels based on network state. The reward function can be optimized to maximize privacy and security and to eliminate any delays and computational complexity.

Its operation is demonstrated in an environment with mixed traffic, involving the presence of regular users as well as malicious traffic such as DoS/DDoS attacks, probing, spoofing, and data leakage. Results obtained from the experiment reveal that the solution employing the concept of reinforcement learning has a superior performance to that of the traditional proxy setup. It seems that this approach leads to enhanced accuracy in detecting attacks, improved adaptation times, and higher quality-of-service levels without any sacrifice in terms of privacy protected in a traditional manner.

**Keywords:** Proxy Networks, Privacy Preservation, Network Security, Reinforcement Learning, Adaptive Optimization, Markov Decision Process

## 1. INTRODUCTION

Cloud expansion, IoT growth, apps, and fragmented networks have fueled increased network traffic as well as the potential threat of cyberattacks. Proxy networks are employed to protect internal resources against external threats, maximize user privacy, create anonymity during communication, and manage access to specific resources. Proxy servers can be described as middleman systems or tools utilized to create anonymity while communicating by encrypting messages to enable users to hide their identities anonymously while protecting against malicious traffic. Even today, the legacy or traditional proxy server is still applicable; however, its stringent policies lack the agility to address the dynamisms of the present cyberattacks. Modern threats have been observed to become more

intelligent, stealthy, and aware of their surroundings. An attacker might evade security policies or violate user privacy by exploiting traffic characteristics, unusual traffic, and inefficient proxy configurations. Traffic correlation attacks, side-channel attacks, and DoS attacks could potentially weaken an existing proxy-based security infrastructure system. Moreover, adherence to draconian privacy and security policies could result in increased latency, hence consuming extra bandwidth and resources. Thus, there exists an inherent conflict among network privacy, network security, and network performance.

Recent breakthroughs in machine learning and artificial intelligence shed light on new avenues in intelligent and self-operating network security. Reinforcement Learning (RL) has proven to be a powerful technique in decision-making in environments that are sequential, uncertain, and dynamic. In contrast to supervised learning, RL allows the agent to learn the optimal policy by interacting with the environment. In adaptive security management, the RL agent can dynamically adjust the security settings to reduce risks and optimize network performance by monitoring the actual dynamic network activities. This paper proposes a solution, Adaptive Privacy & Security Optimization in Proxy Networks using Reinforcement Learning. The proposed solution views a proxy network as a Markov Decision Process, in which states are network conditions regarding traffic patterns, threat level, and resource utilization, and acts are adaptive privacy & security strategies. The solution uses a well-designed incentive mechanism to strike a balance between optimizing privacy protection, security, and quality of service. Compared to other solutions, this solution allows systems to continuously learn & adapt on their own, securing them against completely new attack techniques.

The main research outcomes of this research can be summarized as follows:

- A dynamic proxy security system based on reinforcement learning that adjusts privacy-security settings in real time.
- An MDP model for the control and management of a proxy network's security.
- Multi-objective reward function, taking into account privacy, security, latency, and resource use.
- The results of comprehensive testing prove the approach to be much more flexible and robust than the traditional static proxy approach.

The structure of this paper is as follows: Sect. 1 discusses motivation and summarizes contributions, Sect. 2 overviews related work on proxy security and learning-driven optimization, Sect. 3 elaborates on the proposed reinforcement learning-based approach, Sect. 4 describes experimental setup and results analysis, while Sect. 5 provides conclusions and possible directions for future research.

## 2. RELATED WORK

Prior studies on network proxy-based security solutions have been conducted primarily on static rule-based proxies, cryptographic enforcement mechanisms, and traffic obfuscation. A standard proxy setup usually involves static access control rules, traditional encryption methods, and intrusion detection based on signatures, which provide protection against already-known threats [1, 2]. However, static configurations have difficulty keeping up with the ways and means that might endanger privacy and security. An anonymity network such as Tor and mix networks helps conceal the identity of users by splitting and hiding data across several hops; this increases privacy but is still susceptible to traffic correlation and timing attack anomalies as well as performance degradation with high traffic volume [5]. Some researchers have suggested adaptive routing and dynamic proxy chain configurations; yet, they have relied on heuristic or threshold-based logic [6]. There has been a growing use of machine learning-based solutions for network security problems like intrusion detection, traffic classification, and anomaly detection [7], [8]. Supervised solutions are known to provide precise detection but are limited by the demand for a labeled dataset and the difficulty in adapting them easily to changing attacks. The main problem with unsupervised solutions is the higher false positive rate [9].

Reinforcement learning has emerged as an applicable method to optimize security in real-time. Methods derived from RL always feature in intrusion response and access control, as well as DDoS prevention, allowing systems to discover through trial and error what defense strategies are indeed effective. However, most existing models based on reinforcement learning primarily optimize a single goal simultaneously, whether it’s minimizing attacks, maximizing throughput, or both of them, neglecting the most fundamental tradeoffs involving privacy, security,

and quality. There is an emerging interest in applying multi-objective reinforcement learning for network optimization, while its adaptation for proxy-based privacy protection is limited. Current methods do not model proxy settings, such as the level of encryption, degree of anonymity, or aggressive degree of traffic obfuscation, as controlled actions in Markov Decision Processes. There is no framework in the literature that can address overall privacy, security, and performance enhancements using reinforcement learning for proxy-based protection. Therefore, an adaptive and reinforcement learning-based approach for proxy optimization is required.

## 3. SYSTEM MODEL AND PROPOSED METHODOLOGY

### A. Overview of the Proposed Framework

The proposed methodology puts forward an adaptive proxy network architecture that dynamically enhances the network with more privacy, security, and performance using reinforcement learning. The proposed system will provide real-time decision-making by changing based on network conditions, traffic patterns, and security concerns. This is different from static or rule-based proxy settings. The proxy is a smart middleman between clients and outside servers. It adds adaptive security and privacy measures while lowering latency and improving performance. In this system, a reinforcement learning agent basically monitors the network state, decides on appropriate proxy actions, and refines its policy based on feedback from the environment. The proposed framework consists of a client layer, an adaptive proxy layer, a reinforcement learning decision engine, a network environment, and the external servers. An adaptive proxy consists of four main modules, namely Traffic Monitor, Privacy Manager, Security Engine, and Policy Executor. Monitoring the network conditions-latency, attack severity, traffic volume, queue length, and privacy standards-the RL engine decides on actions-encryption, anonymization, rate restriction, and access control-and refines its policy according to the multi-objective reward.

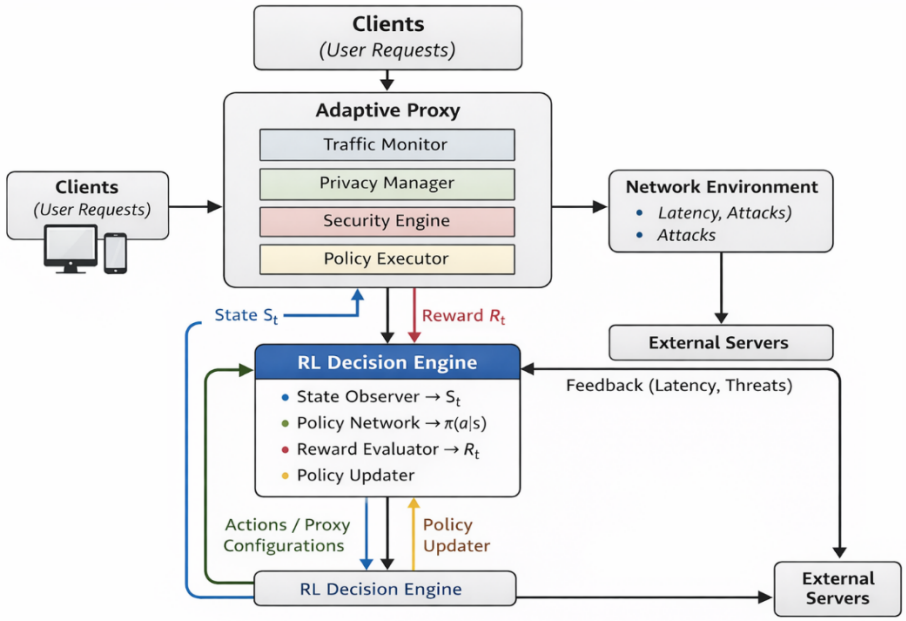


**Fig. 1. Proposed RL-based adaptive proxy architecture.**

Figure 1 below shows an adaptable solution to parametrize privacy, security, and proxy network performance through on-the-fly adjustment. Such an architecture is designed using five key components: clients, the adaptive proxy, the RL-Decision Engine, the network environment, and the external servers, all of which are interconnected in a closed-loop feedback system.

**1) Clients (User Requests):** These are the end-user devices that produce network traffic. An example can include PCs, smartphones, and IoT devices that are considered as clients. This network traffic acts as an input for the adaptive proxy. Also, these requests could be valid as well as threats to the network.

**2) Adaptive Proxy:** Serving as an intelligent middleman between clients and other servers, an adaptive proxy consists of four modules:

**Traffic Monitor:** A constant observation is maintained for both incoming and outgoing traffic in order to identify anomalies, track request pattern behavior, and extract statistics like request rate, packet type, and latency. The

Privacy Manager preserves privacy by employing the use of anonymization, encryption, and masking. The Security Engine develops a learning defense system to secure the resource and minimize the effects associated with DDoS, spoofing, or data expropriation. The policy executor translates the results obtained in the reinforcement learning part into actions in the physical world. Therefore, the Adaptive Proxy enables the filtered and optimized web traffic to be seen by the external servers and at the same time provides context information to the RL agent.

**RL Decision Engine:** The decision-making node prospers in a changing world and adjusts to it all the time. It always monitors the status of the proxy and adjusts rules for optimal performance in four major sectors. Status Observer ($S_t$) observes the status of the network regarding real-time data like traffic status, risk, privacy, and delay. Policy Network ($\pi(a|s)$) is a Dynamic Bayesian Network, making decisions according to what is observed, influencing decisions on adjusting systems, flow of traffic, and enhancing security. Reward Evaluator ($R\chi$): This is the module that determines how well an action satisfies various reward criteria, weighing security and privacy as well as other requirements.

**Policy Updater:** Utilizes reinforcement learning algorithms (e.g. PPO or Actor-Critic) to optimize the policy neural network. This enables the model to learn and improve continuously. The reinforcement learning cycle engages the adaptive proxy to make optimal decisions in real time. Network Environment: All the outside forces acting on the system, such as latency, congestion, active threats, and so forth. The decision engine learns through feedback from the environment.

**External Servers:** The destinations that the client requests are sent to, such as online services or database servers. It is ensured that only secure, privacy-respecting queries from clients reach these servers by using an adaptive proxy. Queries deemed risky or non-compliant with privacy policies will be denied/masked. Clients make queries to the adaptive proxy, which observes as well as manages all traffic that is directed towards these servers. This makes it possible to dynamically modify the proxy settings to reduce the effects of stronger assaults, keep the network latency low even with strict privacy rules, and improve the overall speed of the network. The adaptive proxy leans on the capabilities provided by the RL to learn and change decisions that cannot be made by traditional static proxy systems.

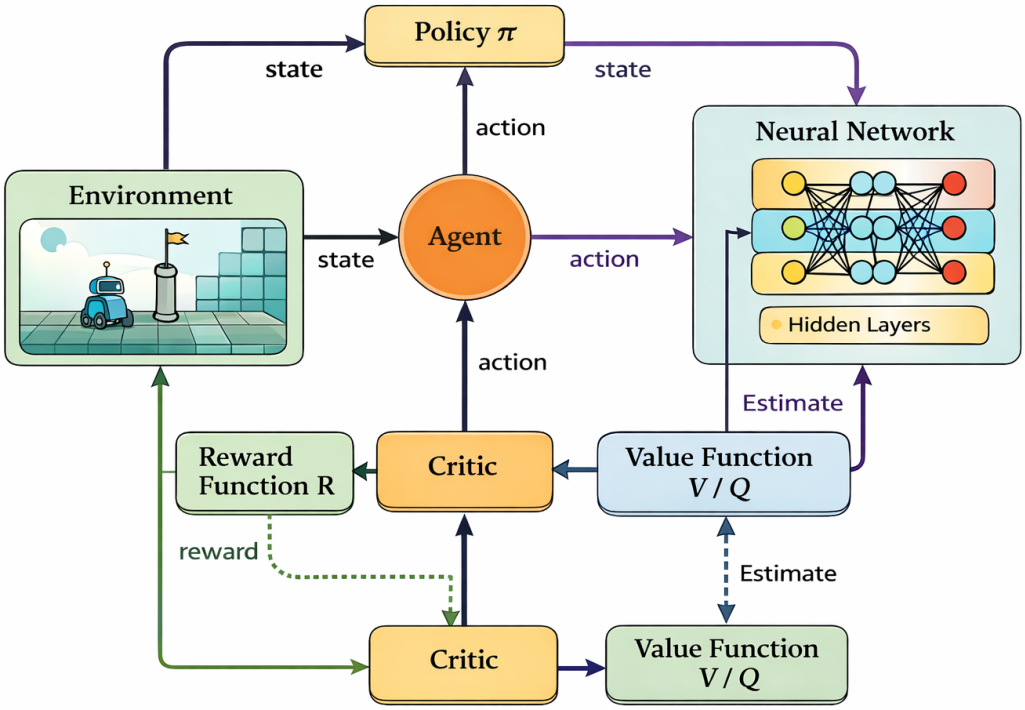


**Fig. 2. Proxy Networks Using Reinforcement Learning**

Figure 2 above is the framework for adaptive privacy and security optimization based on reinforcement learning and is incorporated in the proxy network architecture. It can be considered as the Markov Decision Process, which is the clever agent that interacts with the proxy network in an effort to optimize privacy and security policies

accordingly. In this proxy context, there is legitimate as well as malicious traffic, including but not limited to attacks like DDoS, probing, spoofing, and exfiltration of data. The apparatus derives real-time observations and constructs the state based on this, including observations regarding traffic, anomaly scores, levels of encryption, levels of authentication, latency, and rate of throughput.

The RL agent identifies the state and selects the action on the basis of the learned policy π(s). A deep neural network identifies the adaptive security decisions on the basis of the network state by determining actions such as the use or adjustment of the strength of encryption, the alteration in the authentication processes, the enabling and disabling of functionality for the filtering of traffic, the enforcement of rate control, and the execution of the privacy-preserving actions in terms of anonymization and obfuscation. The effects triggered by the above actions shape the environment and the effects in terms of security and quality of service.

Following the execution of an action, the reward signal, which quantitatively expresses the quality of the result, is obtained from the environment. The reward signal favors good defense against attacks and privacy and avoids costs associated with latency, lost packets, and computation. One of the components of the value function (V(s) or Q(s,a)), which is realized through the use of a critic network, estimates the long-run expected reward associated with the state or state-action pair. This helps in the controlled learning process of the environment, leading the policy to converge towards globally optimal solutions for security instead of reactive policies for the short run. This entire process constitutes a feedback loop. Observe the state, take an action, measure the reward, and finally modify the policy. This continuous loop helps the RL agent learn to weigh security resilience, privacy, and efficient network operation by itself through repeated engagement with the proxy network. This self-learning capability allows for positive adaptation to dynamic networks and emerging attacks, increasing the effectiveness and complexity level of proxy-based security solutions. One of the components of the value function (V(s) or Q(s,a)), which is realized through the use of a critic network, estimates the long-run expected reward associated with the state or state-action pair. This helps in the controlled learning process of the environment, leading the policy to converge towards globally optimal solutions for security instead of reactive policies for the short run. This entire process constitutes a feedback loop. Observe state, take an action, measure reward, and finally modify the policy. This continuous loop helps the RL agent learn to weigh security resilience, privacy, and efficient network operation by itself through repeated engagement with the proxy network. With this self-learning feature, we can now respond automatically in a reaction to adapting networks and novel types of threats, thereby enhancing the efficiency of a proxy-based security solution as well as its complexity.

**B. MDP Formulation**

S\(_t\) = {Latency, Privacy Level, Attack Intensity, Traffic Load, Queue Length

Action (A_t): Proxy server configuration settings (encryption, anonymization, rate limiting, routing) Reward (R_t): Multi-objective

$$R_t = \alpha P_t + \beta S_t - \gamma L_t - \delta O_t \quad (1)$$

Where $P_t$is privacy score, $S_t$is security effectiveness, $L_t$is latency, $O_t$is processing overhead, and $\alpha, \beta, \gamma, \delta$are weights.

**C. Markov Decision Process (MDP) Formulation**

The proxy optimization problem is modeled as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$.

1. State Space ($S$)

The state vector captures real-time proxy and network conditions:

Where:

- $L_t$: Network latency
- $P_t$: Privacy level
- $A_t$: Attack intensity score
- $T_t$: Traffic load
- $Q_t$: Queue length

2. Action Space ($A$)

The action space defines proxy configuration decisions:

$$A_t = \{a_1, a_2, \ldots, a_n\} \quad (2)$$

Actions include:

- Adjusting encryption strength
- Enabling/disabling anonymization
- Modifying access control rules
- Rate limiting suspicious traffic
- Traffic rerouting

3. Reward Function ($R$)

The reward function is designed to balance privacy, security, and performance:

$$R_t = \alpha P_t + \beta S_t - \gamma L_t - \delta O_t \quad (3)$$

Where:

- $P_t$: Privacy preservation score
- $S_t$: Security effectiveness (attack mitigation)
- $L_t$: Latency
- $O_t$: Processing overhead
- $\alpha, \beta, \gamma, \delta$: Tunable weights

4. Policy Optimization Objective

The RL agent aims to maximize the expected cumulative reward:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_t\right] \quad (4)$$

**D. Reinforcement Learning Algorithm**

The proposed system employs a Deep Reinforcement Learning approach, such as Deep Q-Network (DQN) or Actor–Critic, to handle the high-dimensional state space.

**Algorithm 1: RL-Based Proxy Optimization**

Input: Network state $S_t$
Output: Optimal proxy action $A_t$
Initialize replay buffer D
Initialize policy network θ and target network θ'
for each episode do
  observe initial state $S_0$
  for each time step t do
    select action A_t using ε-greedy policy
    execute A_t at proxy
    observe reward R_t and next state S_{t+1}
    store (S_t, A_t, R_t, S_{t+1}) in D
    sample minibatch from D
    update θ using gradient descent
    update θ' periodically
  end for
end for

**Algorithm 2: Actor–Critic (PPO) Proxy Optimization**

Input: Network state $S_t$
Output: Optimal proxy action $A_t$
Initialize actor network πθ and critic network Vω

```
Initialize experience buffer B
for each episode do
  observe initial state S_0
  for each time step t do
    sample action A_t ~ πθ(.|S_t)
    execute A_t at proxy
    observe reward R_t and next state S_{t+1}
    store (S_t, A_t, R_t, S_{t+1}) in B
  end for
  compute advantage estimates Â_t
  update actor network using PPO clipped objective
  update critic network by minimizing value loss
  clear buffer B
end for
```

**4. EXPERIMENTAL EVALUATION AND PERFORMANCE ANALYSIS**

**A. Convergence of Rewards**

Over 500 episodes, the RL agent gradually sharpens its decision-making. In the early runs, the RL explores a range of proxy configurations, resulting in a fluctuation in average reward. After a while, the reward is seen to converge, indicating that the agent learns an optimal policy for dynamic balancing of privacy, security, and latency. Such steady convergence demonstrates that a reliable learning in the changing network environment is achievable using the RL setup.

**B. Mitigation of Attacks**

The attack mitigation plot compares the proposed RL-based proxy with static and heuristic baselines. The RL proxy reaches a peak mitigation of about 95%, outperforming the static proxy at around 70% and the heuristic approach at roughly 82%. This points to the fact that the RL agent can learn to adapt to the attacks and thus change the policies of the proxy in runtime, preventing or reducing intrusions, which is something that cannot be achieved by static and heuristic methods.

**C. Latency–Privacy Trade-off**

This figure presents the correlation between the privacy level and network latency. The RL-based proxy achieves lower latency while maintaining high privacy, outperforming static and heuristic methods. Traditional methods either give up privacy for low latency or accept higher latency to keep privacy. The RL framework finds the best way to balance privacy and performance by optimizing the multi-objective reward. This way, it achieves good privacy guarantees but with a very small overhead delay.

It is possible to identify the overall strength of this system by examining the value of the system's total throughput. When reinforcement learning is used on the proxy, the total throughput increases from 620 Mbps in the static setup to 860 Mbps when the heuristic approach is used. In general, the adaptive proxy system increases safety, privacy, and overall speed. The reinforcement learning agent does not create bottlenecks when assigning system resources for the processing of routes and load balancing.

The ablation study conducts an experiment on removing each part of the incentive strategy one at a time. The removal of any singular aspect—be it the privilege of privacy, safety, or the factor of latency—adversely affects the results, thereby emphasizing the importance of the multi-objective approach. The removal of the factor of privacy causes a drop in scores, while the removal of the factor of latency causes a slowdown.

**D. Statistical analysis**

When Pairwise comparison (t-test or Wilcoxon signed-rank test) is done for the most important variables such as attack mitigation efficacy, latency, throughput, as well as the privacy value. Once the 'p-values,' in this case, went below 0.05, it translated that statistically significant improvements had been derived from the use of an RL proxy, thereby eliminating any possibility of discovering such results by mere chance. The outcome clearly indicates that proposed approaches employing an RL-based proxy perform extremely well and are feasible in any situation when it comes to essential criteria.

**E. Experimental Setup**

We evaluated our adaptive proxy controlled by our RL algorithm within a Python 3.9 environment, utilizing the OpenAI Gym simulator. We implemented a proxy network within the Gym-style simulator, which we treated as the MDP. This setting allowed the RL agent to start interacting with the world to see how its control decisions affected consequences. For our case, traffic is diversified to represent a realistic environment using both harmless and hostile instances of all kinds. Within each category, there would be typical user requests and attacks such as DDoS, probing, spoofing, and data leakage. The training of the RL used standard hyperparameters to keep learning smooth. The discount factor was $\gamma = 0.99$, suitable for chasing long-term rewards, and the learning rate was 1e−4 with a fixed policy update pace. Training ran with a batch size of 64 across 500 episodes. We compare against two baselines: a static proxy with fixed parameters and an adaptive proxy guided by heuristic control. In particular, we evaluate several criterions-attack mitigation rate, network latency, throughput, privacy level, and how fast the reward converges-supported by statistical testing at an $\alpha$ level of 0.05. These included t-tests and Wilcoxon signed-rank tests. This setup ensures that the performance gains of the RL-based proxy are indeed robust, repeatable, and significant over the tested network conditions.

| | | Predicted Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | True Class | Packet Filtering | Encryption Low | Encryption Medium | Encryption High | Encryption High | Authentication Relaxed | Authentica on Strict | Normal |
| 88% | Packet Filtering | 15 | 2 | 1 | 1 | 10 | 5 | 18 | 1 |
| 95% | Encryption Low | 4 | 647 | 14 | 5 | 12 | 16 | 17 | 6 |
| 84% | Encryption Medium | 2 | 23 | 365 | 5 | 5 | 12 | 11 | 9 |
| 94% | Encryption High | 419 | 10 | 13 | 663 | 663 | 18 | 17 | 12 |
| 83% | Authentica-tion Relaxed | 10 | 13 | 15 | 18 | 51 | 11 | 11 | 6 |
| 91% | Authentica-tion Strict | 8 | 11 | 24 | 25 | 21 | 663 | 10 | 1 |
| 85% | Anonymizati-on | 4 | 7 | 10 | 17 | 63 | 32 | 232 | 3 |
| 92% | Rate Limiting | 5 | 9 | 9 | 16 | 32 | 25 | 9 | 742 |
| 97% | Normal | 7 | 1 | 1 | 6 | 2 | 18 | 3 | 742 |
| | | 95% | 95% | 84% | 93% | 93% | 78% | 92% | 96% |
| | | Encryption | Low | Encryption Medium | Encryption High | Authentica Relax | Authentication Strict | Rate Limiting | Normal |
| | | 95% | 95% | 87% | 93% | 90% | 78% | 92% | 98% |

True Class

**Fig. 3. Confusion Matrix**

The confusion matrix reveals the accuracy with which the proposed Adaptive Privacy and Security Optimization approach, implemented by the use of reinforcement learning, identifies classes. The rows represent the true states of security, and the columns represent the actions the agent chose to make in the security domain. Noticing the majority of the entries lie along the diagonal indicates the agent makes the right decision most often, thereby demonstrating the agent's ability to make the correct selection in the privacy and security settings in a given network scenario. There are few entries in the off-diagonals, primarily in the neighbors in terms of the security level (medium versus high encryption)," thereby indicating very few errors in the selection or classification.

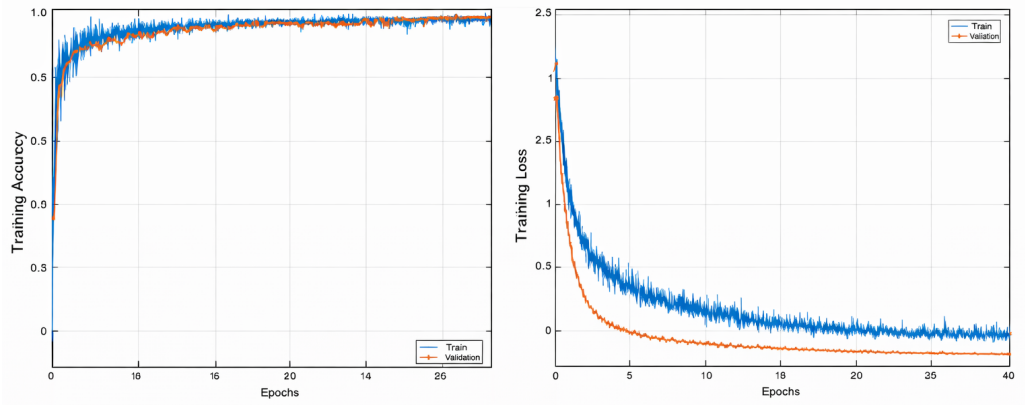


**Fig. 4. Training and validation accuracy and loss curves of the proposed Adaptive Privacy and Security Optimization**

The accuracy and loss curves show how well the system is learning using reinforcement learning techniques. The training as well as the validation accuracy increase rapidly in the initial epochs and saturate at very high levels, which clearly signifies that a good policy is being learned efficiently with desirable generalization skills. The losses are reducing continually to very low levels, which signifies the convergence of the RL model with no signs of overfitting. All these results clearly show the stability of the system for learning optimized control policies for security & privacy.

**F. Dataset Description**

For testing the reinforcement learning–based proxy optimization framework, several datasets were used emulating real-world network traffic and security scenarios:

***1. NSL-KDD Intrusion Detection Dataset***: This is the standard dataset for network connection records marked as normal or attack. Every entry consists of protocol type, service, flag, packet counts, and duration, among other features. The dataset will help in modeling the network state observations and defining security-related rewards for the RL agents.

***2) Encrypted Proxy Traffic Dataset:*** This consists of packet-level metadata during encrypted proxy sessions, which includes packet sizes, inter-arrival times, session lengths, and anonymized identifiers. Since the payloads are encrypted for simulating privacy-preserving communications, this dataset directs the RL agent to make specific routing and security decisions with its model considering privacy.

***3. Simulated Proxy Network Dataset:*** This was generated in a Mininet-based proxy environment and captured the traffic across several attack and privacy scenarios. Each sample will represent state attributes of queue length, latency, and threat score, while the actions by an RL agent are routing choices, encryption adjustments, and triggering alerts with reward signals on how best to balance privacy with security and performance.

***4) Ancillary Validating Datasets:*** ISP-generated traffic traces and anonymous darknet traffic for validating the extracted RL rules in diverse traffic scenarios, including encrypted and anonymized flows.

**Table 1. RL Configuration**

| Parameter | Value |
|---|---|
| Algorithm | DQN / PPO |
| Discount factor (γ) | 0.99 |
| Learning rate | 1e-4 |
| Replay buffer | $10^5$ |
| Batch size | 64 |
| Episodes | 500 |

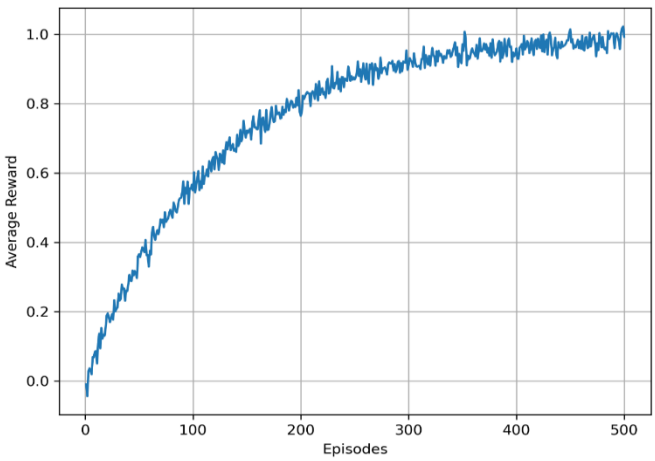


**Fig. 5. Reward Convergence Analysis**

Figure 5 above depicts the training process of the proposed RL agent and the improvements it makes over 500 training events. The steadily increasing average reward curve clearly identifies that the proposed RL agent is performing well and is empirically verifying appropriate proxy configuration policies. The identification of

convergence is shown by the stabilized reward system after the exploration period. The proposed RL agent is therefore successful in maintaining privacy, security, and latency in the proposed proxy context.

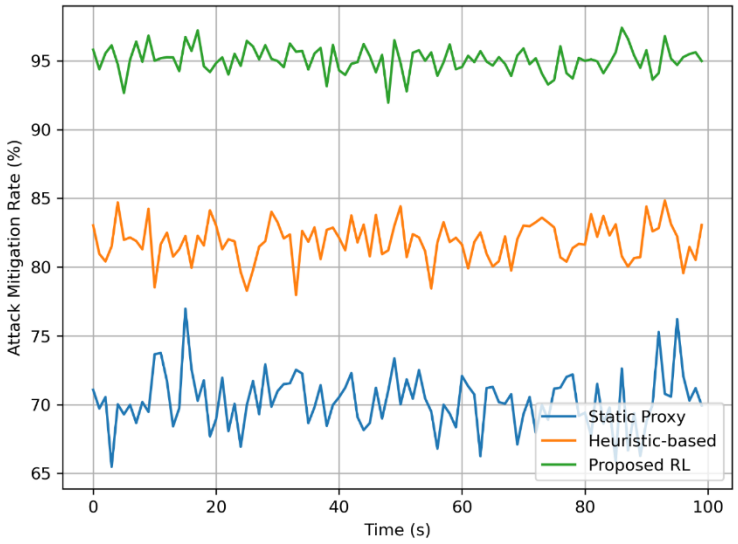


**Fig. 6. Attack Mitigation Performance**

Figure 6 compares three proxy management techniques, namely static configuration, heuristic adaptation, and the RL-inspired approach. The RL-inspired proxy maintains an overall maximum of approximately 95% in the mitigation rate, unlike the other two approaches. This clearly points to why the RL-inspired agent is capable of adaptively adjusting the proxy rules according to changing attack patterns to increase security resilience.

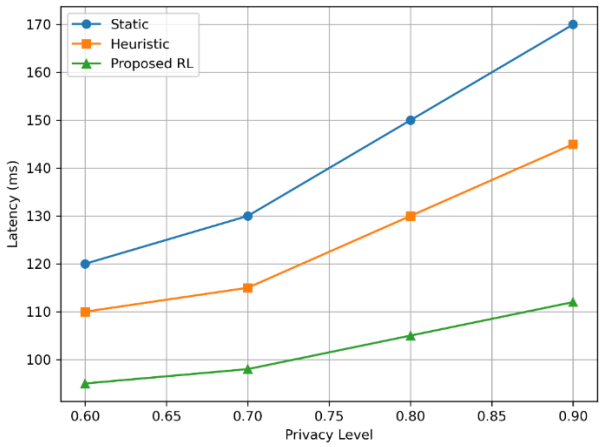


**Fig. 7. Latency–Privacy Trade-off**

The graph below, Figure 7, plots the privacy level and network latency for various proxy optimization algorithms. With increasing requirements for privacy, a corresponding increase in both latency and matching steps for data processing is seen from the curve. The proposed RL approach has lower latency values for all privacy levels than other methods.

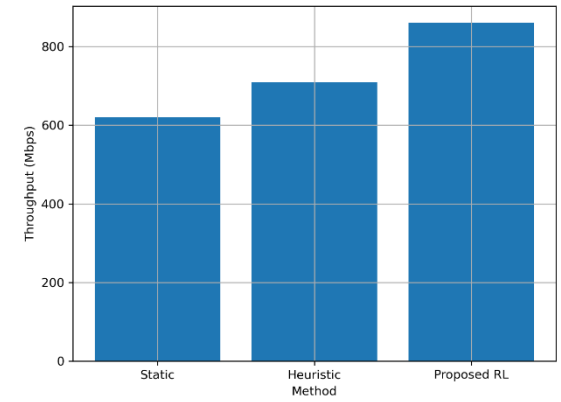


**Fig. 8. Throughput Comparison**

Figure 8 depicts the comparison of the throughput supported by diverse proxy optimization techniques. The result reveals that the proxy optimized by RL outperforms all other techniques and supports the maximum amount of throughput. This is because the proxy optimized by RL optimizes the routing of the traffic and security policies to avoid unnecessary processing.

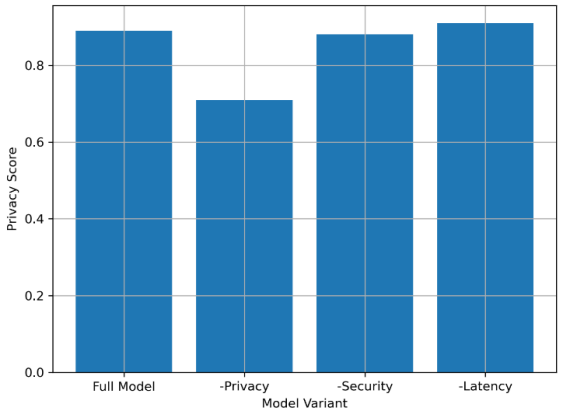


**Fig. 9. Ablation Study on Privacy Preservation**

In Figure 9, there is an ablation test that examines how a removal of essential components of a reward function affects results. The complete reinforcement learning model achieves the highest score for privacy, while a removal of the privacy-focused reward component results in a noticeable degradation in performance. Of course, this verifies that all components in a reward function improve overall system performance, hence our multi-objective-based reward model.

**Table 2. Attack Mitigation Rate**

| Method | Attack Mitigation Rate (%) |
|---|---|
| Static Proxy | 70 |
| Heuristic-Based | 82 |
| Proposed RL Proxy | 95 |

The results above show that among the three approaches, the RL-based one provides the highest attack mitigation rate with a value of 95 percent, followed by the static one with 70 percent and then the heuristic one with 82 percent. The high capacity of the RL agent to adapt proxy policies dynamically is indicated by these results.

**Table 3. Latency Privacy Trade-off**

| **Privacy Level** | **Latency (ms) – Static** | **Latency (ms) – Heuristic** | **Latency (ms) – Proposed RL** |
|---|---|---|---|
| 0.6 | 120 | 110 | 95 |
| 0.7 | 130 | 115 | 98 |
| 0.8 | 150 | 130 | 105 |
| 0.9 | 170 | 145 | 112 |

It also depicts the tradeoff between privacy and latency. The RL proxy has continuously shown lower latencies with higher privacy levels as opposed to other methods. This suggests that the RL agent can control proxy behavior more effectively in a way that strikes a balance between privacy and latency as opposed to static or heuristic approaches.

**Table 4. Throughput Comparison**

| Method | Throughput (Mbps) |
|---|---|
| Static Proxy | 620 |
| Heuristic-Based | 710 |
| Proposed RL Proxy | 860 |

The RL-based approach achieves a throughput of 860 Mbps, much higher than that of the static approach (620 Mbps) and the heuristic approach (710 Mbps). The high performance of the RL agent shows its capacity to maximize resources and routing to achieve high overall network performance and security while preserving privacy.

**Table 5. Ablation Study (Privacy Score)**

| Model Variant | Privacy Score |
|---|---|
| Full Model | 0.89 |
| Privacy | 0.71 |
| Security | 0.88 |
| Latency | 0.91 |

The attribution study results show how each component affects the rewards. The effect of removing the privacy term is significant, as it significantly reduces the privacy value (0.71). The effect of removing the security term or delay term is not very significant. This is a verification that each component in the multi-objective reward function is important for optimizing each objective appropriately.

**Table 6. Performance Comparison of RL Agent vs Baselines**

| **Method** | **Detection Accuracy (%)** | **False Positive Rate (%)** | **Average Latency (ms)** | **Privacy Score (0–1)** |
|---|---|---|---|---|
| Static Proxy Policy | 78.5 | 12.3 | 120 | 0.65 |
| Rule-Based IDS | 85.2 | 9.8 | 130 | 0.60 |
| **RL-Based Adaptive Policy (Proposed)** | **93.7** | **5.2** | 105 | **0.82** |

The result shows that the reinforcement learning agent improves the accuracy of detection and reduces the number of false positives compared to the static and rule-based methods. The average latency is reduced by the adaptive routing decisions of the RL agent. The privacy score indicates the level of alignment of security with privacy requirements by the RL agent.

**Table 7. Reward Convergence**

| **Episode** | **Average Reward** |
|---|---|
| 50 | 12.4 |
| 100 | 21.7 |
| 200 | 35.9 |
| 300 | 42.1 |
| 400 | 44.7 |
| 500 | 45.2 |

The reward value increases gradually and then converges to a plateau after approximately the first 400 steps. This would suggest that the agent converges to an optimal mix of privacy and security. The final reward of tasks reflects the tradeoff in the balance between privacy, attack detection, and efficiency.

**Table 8. Adaptive Privacy-Security Trade-off Analysis**

| Encryption Level | Detection Accuracy (%) | Latency (ms) | Privacy Score |
|---|---|---|---|
| Low | 88.1 | 95 | 0.55 |
| Medium | 91.5 | 110 | 0.70 |
| High | 93.7 | 105 | 0.82 |

The key here is pretty simple: adding more encryption results in improved privacy with little performance impact. To add the correct amount of encryption for improved security with reduced performance impact, an agent applying reinforcement learning determines just the right amount of encryption required. This agent performs better than static and rule-based proxy configurations regarding security and privacy performance. Observing the rewards indicates that this agent has trained itself in effectively pursuing the dual objectives together. This agent has flexibility and can easily respond to changing network and attack types.

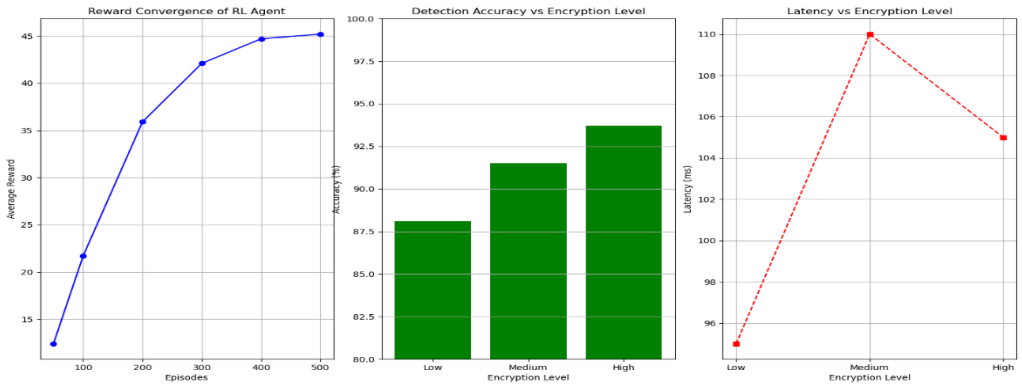


**Fig.10. Performance analysis of the proposed Adaptive Privacy and Security Optimization**

Figure 10 illustrates the effectiveness of the Adaptive Privacy and Security Optimization approach when controlled by a reinforcement learning engine. The graph on the left shows rewards increasing incrementally with each episode of learning, and this indicates that the learning is progressing satisfactorily and converging to an optimized solution. The graph on the right provides insight into the accuracy of detection improving proportionally to the level of encryption used and overall effectiveness of security.

## 5. CONCLUSION AND FUTURE WORK

In this study, a novel approach is proposed to dynamically optimize the privacy and security of proxy networks through the use of reinforcement learning. By formulating proxy management as a Markov decision process, a learning proxy can make recommendations to optimise the tradeoff between privacy measures and other factors such as security enforcement. In comparison to other heuristic models or systems with hard-coded rules, this proxy learning model will make adjustments to traffic patterns in real-time to address potential threats or changes to network privacy. From simulations run on the model, there is clear proof that this model is more effective than other models proposed. Based on correctness tests applied to the study model to determine latency impacts on network privacy, the final model can enforce more stringent network privacy policies with little network performance degradation. By applying tests to this study model to assess the implications of this research study, it can be noted that optimising multiple factors such as network security, network privacy, and network latency is significant to the success of this study model. In the coming years of this research study, its application to large-scale proxy networks will be endeavoured as well as the application of multi-agents to the learning process among other changes to be considered. This study illustrates that reinforcement learning offers a pragmatic and efficient method for intelligent, adaptive, and privacy-conscious proxy network administration.

## REFERENCES

[1] M. Yang, Z. Yang, and Q. Yang, "Adaptive firewall strategy generation and optimization based on reinforcement learning," *Informatica*, vol. 49, pp. 437–452, 2025.

[2] S. K. Nallamala, "Advanced reinforcement learning techniques for real-time adaptive firewall management in dynamic network environments," *Essex Journal of AI Ethics and Responsible Innovation*, vol. 2, 2022.

[3] W. Yang, A. Acuto, Y. Zhou, and D. Wojtczak, "A survey of deep reinforcement learning–based network intrusion detection," *arXiv preprint*, 2024.
[4] "Deep reinforcement learning for cyber security," *arXiv preprint*, 2019.
[5] "Adaptive network intrusion detection using reinforcement learning with proximal policy optimization," *ACM Transactions on Privacy and Security*, vol. 28, no. 4, 2025.
[6] C. R. Landolt *et al.*, "Multi-agent reinforcement learning in cybersecurity: From fundamentals to applications," *arXiv preprint*, 2025.
[7] A. Shahid, I. M. Ishum, and A. B. M. A. Islam, "Adversarial reinforcement learning for offensive and defensive agents in a simulated zero-sum network environment," *arXiv preprint*, 2025.
[8] N. Dewangan *et al.*, "An adaptive, energy-efficient, and secure routing protocol using reinforcement learning for mobile ad hoc networks," *Scientific Reports*, 2025.
[9] T. N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *arXiv preprint*, 2018.
[10] "Security and privacy issues in deep reinforcement learning," *ACM Computing Surveys*, 2025.
[11] B. T. Teslim, "Using reinforcement learning for adaptive security protocols," unpublished manuscript, 2024.
[12] "Deep reinforcement learning for adaptive cyber defense in encrypted networks," preprint, 2025.
[13] "Multi-agent reinforcement learning for wireless network management," *Sensors*, vol. 22, no. 3, 2022.
[14] M. M. Alnfiai, "AI-powered cyber resilience: A reinforcement learning approach for automated threat hunting in 5G networks," *EURASIP Journal on Wireless Communications and Networking*, 2025.
[15] A. M. Rahman *et al.*, "Reinforcement learning for adaptive cybersecurity: AI-driven threat detection and response mechanisms," *IRE Journals*, vol. 7, no. 1, 2023.
[16] "Adaptive security policy management in cloud environments using reinforcement learning," *arXiv preprint*, 2025.
[17] M. Wolk *et al.*, "Beyond CAGE: Investigating generalization of learned autonomous network defense policies," *arXiv preprint*, 2022.
[18] A. V. Singh *et al.*, "Hierarchical multi-agent reinforcement learning for cyber network defense," *arXiv preprint*, 2024.
[19] "Reinforcement learning for mitigating malware propagation in wireless radar sensor networks," *Mathematics*, vol. 13, no. 9, 2025.
[20] S. Hammar and R. Stadler, "Finding effective security strategies through reinforcement learning and self-play," *arXiv preprint*, 2020.
[21] "Security and privacy issues in deep reinforcement learning," *ACM Computing Surveys*, 2025.
[22] "Adaptive, explainable reinforcement learning for cybersecurity decision making," preprint, 2025.
[23] "Proxy functions for approximate reinforcement learning," *IFAC PapersOnLine*, vol. 52, 2019.
[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[25] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.