ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

Reinforcement Learning for Continuous Cyber Threat Detection Rule Improvement

Nabeela Temitayo Adebola[1]; Williams Ezebuilo Eze[2]; Kamoru Emmanuel Umoru[3]; Jamiu Akande[4]; Nuhu Ezra[4]

[1]Department of Cybersecurity, University of Salford, Salford, England.
[2]Engineering Team, Rolla Finance, California, USA.
[3]Department of Applied Artificial Intelligence and Data Analytics, University of Bradford, Bradford, UK.
[4]Center for Cyberspace Studies, Nasarawa State University, Keffi, Nigeria.

Publication Date: 2026/03/13

Abstract: Security Information and Event Management systems are still at the core of threat monitoring in enterprises, but their rule-based detection methodology is mostly static in nature and always in need of manual tuning. As the nature of cyber-attacks becomes increasingly sophisticated, the complexity of the operating environment also increases, leading to a degradation in the accuracy of the rule-based methodology, with false positives and false negatives rising significantly. Recent studies show that adaptive learning methodologies can improve the accuracy of anomaly-based detection systems in a dynamic operating environment. Reinforcement learning is a methodology in which a learning agent learns through its interactions with its operating environment and improves its decision-making capabilities through a series of iterations. This research proposes a reinforcement learning-based framework for the continuous improvement of cyber threat detection rules in Security Information and Event Management systems. A reinforcement learning agent learns from the outcomes of the alerts generated in the system, the feedback from the Security Operations Center, and the threat intelligence available in the system to improve the thresholds and correlation values in real time for the rule-based methodology. This research uses benchmark intrusion datasets to evaluate the proposed methodology and compares its performance with static rule-based systems to show the improvement in accuracy and a reduction in false positives generated in the system.

Keywords: *Reinforcement Learning, SIEM, Continuous Monitoring, False Positives, SOC Automation, Adaptive Cyber Defense*.

How to Cite: Nabeela Temitayo Adebola; Williams Ezebuilo Eze; Kamoru Emmanuel Umoru; Jamiu Akande; Nuhu Ezra (2026) Reinforcement Learning for Continuous Cyber Threat Detection Rule Improvement. *International Journal of Innovative Science and Research Technology*, 11(3), 486-495. https://doi.org/10.38124/ijisrt/26mar324

I. INTRODUCTION

➢ *Background of the Study*

However, in modern days, the ever-evolving cyber threat landscape, including APT, polymorphic malware, and automated exploitation frameworks, makes the situation a fluid battleground for the defenders. Security Information and Event Management systems collect logs, correlate events, and trigger alerts based on predetermined rules, which are normally based on expert knowledge. However, these rules are based on fixed thresholds and signature-based logic. This type of logic performs well in detecting known attack patterns but does not perform well in detecting new or evolving attack patterns. Traditional intrusion detection systems and correlation systems require periodic hand-tuning to keep them performing well, and as discussed in [1], the static nature of intrusion detection systems can change with the data set. This change causes a phenomenon known as concept drift, which increases false positives and false negatives, adversely affecting system efficiency and confidence in the system for analysts. To add to this, alert fatigue, which affects SOC analysts, also plays a role in this problem statement because these analysts are required to respond to a flood of alerts within a short time frame. Reinforcement learning is a branch of machine learning that gives a mathematical basis for making sequential decisions under uncertainty [3]. This type of machine learning has been explored for its application in cybersecurity in adaptive defense, automated responses, and anomaly detection [2], [4]. However, its use in continuous rule-tuning for SIEM systems has not been explored in detail.

➢ *Problem Statement*

It is a known fact that modern SIEM systems are based on static or manually tuned rules for detecting cyber threats. However, these rules are not reliable in the long run because cyber attackers are evolving with time, environments are changing, and user behavior is also changing. Moreover, false positives are also a problem in these systems, leading to alert fatigue and false negatives, which are not desirable in a system. Therefore, what is required is a system that adapts and refines its own rules based on real-time feedback from

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

the system, and without such a system, the current monitoring systems are not reliable for detecting cyber threats.

➢ *Aim and Objectives*

This study takes a reinforcement learning-based approach to optimizing cyber threat detection rules within SIEM environments. The objectives of this study are:

- To develop a reinforcement learning-based approach to optimizing SIEM rules, where the objectives, actions, and rewards are clearly defined.
- To develop a reward function that strikes a balance between optimizing detection accuracy and minimizing false positives.
- To incorporate feedback from SOC analysts and real-world alerting outcomes to guide continuous learning.
- To compare and validate the proposed reinforcement learning-based approach to optimizing SIEM rules with traditional rule-based approaches using standard metrics for comparison.

➢ *Research Questions*

- How can SIEM rule optimization be most effectively achieved through a reinforcement learning-based approach?
- What is the most effective way to balance detection accuracy and false positives in a reinforcement learning-based approach to SIEM optimization?
- To what extent can a reinforcement learning-based approach to SIEM optimization help reduce false positives and false negatives?

➢ *Significance of the Study*

This study is significant because it helps improve adaptive cybersecurity by providing a framework for optimizing SIEM rules within a live environment. This is done by incorporating reinforcement learning, which helps improve accuracy and reduces manual intervention, thereby increasing SOC analyst productivity. This is a significant study because it provides a practical application and validation of reinforcement learning within a real-world environment.

➢ *Scope of the Study*

This is a study focused on optimizing SIEM rules using reinforcement learning, where the results are evaluated using benchmark intrusion datasets and simulated SOC feedback. This is not a study focused on automating response orchestration and endpoint remediation strategies.

II. LITERATURE REVIEW

➢ *Traditional Security Information and Event Management and Rule-Based Detection Systems*

Security Information and Event Management systems collect log data from a variety of sources and use correlation rules to identify suspicious activity. These rules are often based on expert knowledge and trends in known attack activity. Traditional intrusion detection systems are broadly classified into two types: signature-based systems and anomaly-based systems [5]. Signature-based systems use known patterns to identify known attack activity, while anomaly-based systems use known normal activity to identify unusual activity. One of the biggest disadvantages of rule-based systems is their inability to adapt to changing attack methods. As attacker methods change, the rules used in the intrusion detection system will become less effective over time. Concept drift, or the change in the statistics of network traffic, will also limit the effectiveness of these systems [1].

➢ *Machine Learning in Intrusion Detection*

Machine learning has been brought in to increase the accuracy of intrusion detection systems and improve their ability to recognize patterns in network traffic. Supervised learning algorithms such as support vector machines, decision trees, and neural networks have been shown to increase classification accuracy using benchmark data sets [6]. Deep learning techniques, using recurrent and convolutional neural networks, improve the ability to recognize patterns in network traffic [7]. However, these supervised learning algorithms still require labeled data sets, which are often not available in real-time due to the time required for verification Additionally, these systems are still static in nature and will not adapt well to changing attack methods [1]. To overcome the problem of not having labeled data sets, researchers have suggested using unsupervised learning algorithms for intrusion detection. However, these algorithms often cannot distinguish between normal and abnormal activity, leaving many false alarms in production systems.

➢ *Reinforcement Learning Foundations*

Reinforcement learning explores means to make decisions in steps in a changing world. An agent observes the world, takes actions, and obtains rewards to influence the way the agent's policy improves [3]. It doesn't use labeled data like supervised learning; instead, it aims to maximize rewards in the long term. The basis of reinforcement learning is the Markov Decision Process, which is the mathematical basis of the aforementioned concepts [9]. Q-learning, policy gradients, and actor-critic methods are the techniques to find the optimal policy even in uncertain situations [3]. Recently, deep reinforcement learning uses neural networks with reinforcement learning optimization to cope with high-dimensional state spaces. Given the ability to cope with uncertain situations, reinforcement learning is a good fit in cybersecurity monitoring systems [8].

➢ *Reinforcement Learning in Cybersecurity*

Research has also been done to investigate the potential of reinforcement learning in improving intrusion detection and automated defense. Aref et al. surveyed various intrusion detection methods using reinforcement learning. They observed the effectiveness of adaptive threat recognition, which enhances the quality of intrusion detection compared to static methods [2]. Lin et al. also highlighted the potential of using deep reinforcement learning to optimally improve cyber defense strategies [4]. In reinforcement learning-based network defense, scenarios such as dynamic firewall configuration, moving target defense, and automated

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

response are explored [4]. In these scenarios, the reinforcement learning agent learns the optimal move to make in defense against attackers based on simulated attacker behaviors. In the research done by Apruzzese et al., the potential of reinforcement learning in addressing the problem of adversarial drift in intrusion detection systems was highlighted [1]. They emphasized the importance of adapting the intrusion detection system to the changing behaviors of attackers. Reinforcement learning allows the updating of the policy to address the adversarial drift. However, the majority of the research done using reinforcement learning in the context of cybersecurity has been focused on the response mechanisms. Very little research has been done to optimize the parameters of the SIEM system using the feedback from the SOC analyst.

➢ *Reward Function Design in Security Contexts*

One of the biggest challenges in applying reinforcement learning to cybersecurity is the reward function. A bad reward function can cause the model to learn the wrong thing. In detection problems, it's a balancing act. We want to maximize true positives. We also want to be careful about how we handle false positives and negatives. There is research suggesting the use of multi-objective optimization techniques that consider factors like detection accuracy, resource usage, and overall efficiency [4]. In SIEM systems, for example, alert fatigue isn't just annoying – it can also be a cost factor in the reward function. As Sutton and Barto note, the reward function should be aligned with the long-term goals [3]. To translate this into a cybersecurity context, there is a need to incorporate feedback from analysts, results from confirmed incidents, and the latest intelligence.

➢ *Research Gaps*

From the literature, it is clear that significant advancements have been made in machine learning-based intrusion detection and, more importantly, reinforcement learning-based solutions for cybersecurity. Nevertheless, it is apparent that there is a need to address the following challenges:

- Current SIEM solutions are heavily rule-based, and although there is some form of adaptive automation, it is limited.
- Current cybersecurity reinforcement learning solutions focus more on automating response, not optimizing rules.
- Only a handful of studies have successfully incorporated feedback from SOC analysts directly into the reinforcement learning objective function.
- There is a lack of empirical comparison between reinforcement learning-based rule optimization and traditional SIEM solutions. To address these challenges, it is necessary to develop a framework where SIEM rule optimization is a reinforcement learning problem, feedback is incorporated, and it is quantifiable.

III. METHODOLOGY

➢ *System Architecture*

The system design introduces the reinforcement learning agent into the SIEM system to facilitate adaptive tuning of the detection rules. The agent observes the outcome of the alerts, the system context in which the alerts are generated, and the feedback from the human analysts before making appropriate changes to the parameters of the detection rules.

- *Architectural Overview*

The architecture of the system can be described as follows:

✓ Log and Event Sources
✓ SIEM Correlation Engine
✓ Detection Rule Parameter Layer
✓ Reinforcement Learning Agent
✓ SOC Feedback Interface
✓ Reward Computation Module
✓ Performance Monitoring and Evaluation Unit

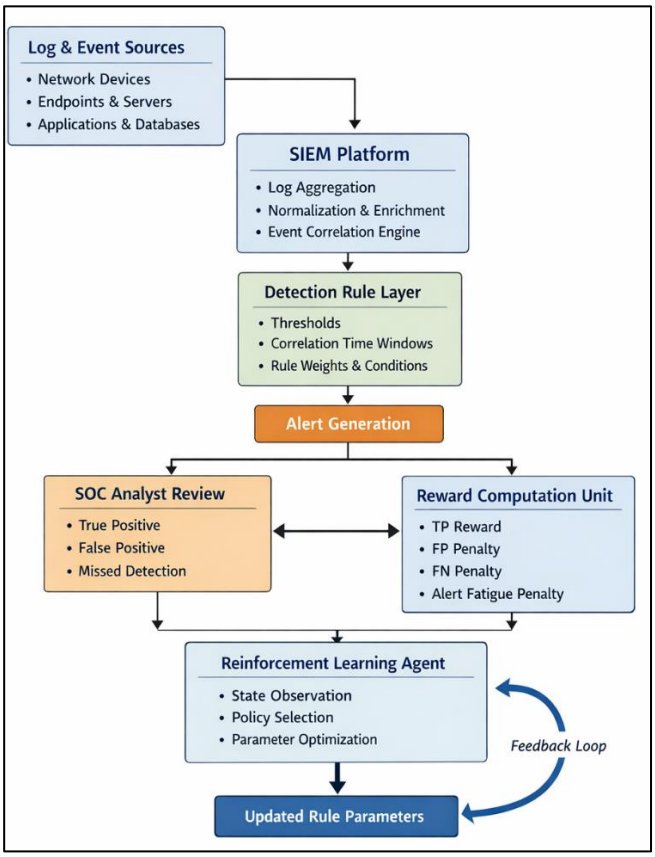

Fig 1 Proposed RL-Driven SIEM Architecture

The architecture illustrates a self-adjusting and closed-loop architecture. The logs and streams of events enter the SIEM platform and then exit as alerts after being processed by the correlation rules. The SOC analysts examine the alerts simultaneously with the processing of the alerts by the reward computation unit. Based on the detection results, the reward signals enter the reinforcement learning (RL) agent. The RL agent uses the information about the current system state to adjust the parameters of the detection rules. The adjusted parameters enter the rule layer and create a loop of continuous adaptation of the detection parameters based on real-world outcomes.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

➢ *Problem Formulation as a Markov Decision Process*

Tackling SIEM rule optimization can be viewed as a Markov Decision Process (MDP), a widely used model in reinforcement learning to make sequential decisions in uncertain environments. In this context, the system can be characterized by the 5-tuple ((S, A, R, P, γ)), in which S represents the state space, A represents the action space, R represents the reward function, P represents the state transition probability, and γ represents the discount factor, balancing current and future rewards. By modeling the SIEM rule optimization problem as a Markov Decision Process, a reinforcement learning agent can explore rule modifications and learn optimal policies to improve system performance.

- *State Space (S)*

The state space represents all the necessary information to make a decision on how to modify the rules by a reinforcement learning agent. Each state is a set of features, each of which is a component of the state vector, given by:

✓ Current rule thresholds and weights: numeric values to trigger alarms, e.g., minimum events, confidence weights, or correlation thresholds.
✓ Alert statistics: key performance indicators, e.g., true positives, false positives, or false negatives in a certain time window.
✓ Traffic distribution: overall traffic statistics, e.g., volumes of packets, rates of connections, or frequencies of log events.
✓ SOC analyst validation: feedback from Security Operations Center analysts, e.g., validation of alerts as being correct or incorrect.
✓ Contextual threat intelligence: external or internal intelligence signals, e.g., threat intelligence about emerging threats, known vulnerabilities, or threat severity.

To avoid any potential bias in learning, all components of the state vector are normalized before being fed to the reinforcement learning agent.

- *Action Space (A)*

The action space is essentially an overview of all the things the reinforcement learning agent can change about the rules of detection. The different possible moves the agent can make include: Increasing or decreasing the thresholds of the rules, Modifying the correlation time windows, which affects the grouping and correlation of the detected items, Modifying the confidence weights of the rules based on the level of importance the conditions being detected have and turning some of the rule conditions on or off based on their performance. In the case of tabular Q-learning, the actions are discretized, meaning the agent can choose from a variety of tweaks in the parameters. However, in the case of policy gradient methods, the agent is able to make continuous changes in the rule parameters.

- *Reward Function (R)*

The reward function is a measure of how well the agent behaves, considering the effectiveness of the detection of threats and the efficiency of the operation of the system. It is designed to direct the agent to rule setups in which the actual detection of threats is maximized, together with keeping false alarms in check. The reward function Rt at any time t is given by the equation:

$$Rt = \alpha \cdot TP - \beta \cdot FP - \gamma \cdot FN - \delta \cdot AF \text{ ---------- (1)}$$

- *Definitions:*

✓ TP: True Positives, or the detection of actual security threats, which adds to the positive reward function.
✓ FP: False Positives, or threats detected as actual threats, which are penalized to discourage the generation of too many alarms.
✓ FN: False Negatives, or threats missed by the system, which are heavily penalized since threats are actual security concerns.
✓ AF: Alert Fatigue Penalty, or a penalty for too many alarms, or alarms generated by low-value threats. The parameters α, β, γ, and δ can be set to any value, enabling you to adjust the importance of each component of the equation according to your needs and priorities in the organization. For example, a higher value of γ can be set to stress the importance of not missing threats, and a higher value of δ can be set to discourage too many alarms.

Table 1 Reward Function Parameter Weights

Parameter	Description	Weight Symbol	Impact on Learning
True Positive	Correct detection	α	Positive reward
False Positive	Incorrect alert	β	Penalty
False Negative	Missed attack	γ	Strong penalty
Alert Fatigue	Excessive alert volume	δ	Operational penalty

➢ *Reinforcement Learning Algorithm Selection*

Two reinforcement learning algorithms are implemented for comparative evaluation:

- Q-Learning
- Deep Q-Network (DQN)

Q-learning provides interpretability and stability for discrete action spaces. DQN is used to handle high-dimensional state representations using neural network approximators [8].

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

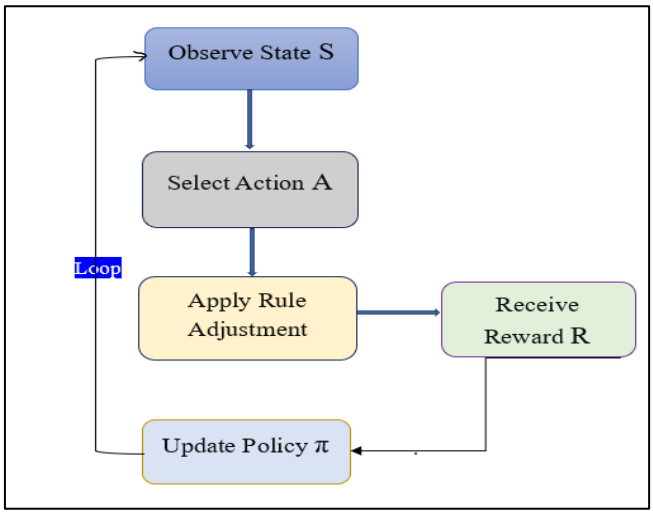

Fig 2 RL Learning Cycle

Figure 2 shows the reinforcement learning process for SIEM rule optimization in a repeating cycle. This begins with "Observe State" or "S," where the RL agent observes the current system state. This is a vector representing the current rule parameter values, alert statistics like true positives, false positives, false negatives, traffic distribution metrics, SOC analyst validation results, and threat intelligence indicators. Normalization is applied for stability in the learning process. Next is "Select Action" or "A," where the agent determines the specific actions to take in tweaking the SIEM rules. This could include adjusting thresholds, changing time windows in the correlation engine, adjusting confidence weights for each rule, or enabling/disabling specific rules. This is a discrete action for the Q-learning agent or a continuous action for a policy gradient agent. "Apply Rule Adjustment" is the next step in the cycle, where the agent applies the selected rule parameter changes in the SIEM correlation engine. This immediately affects the operation of the rules in processing new events and generating alerts. "Receive Reward" or "R" is the step where the system evaluates the alert outcomes to quantify the quality of the agent's actions. True positives are rewarded, while false positives, false negatives, and alert fatigue are penalized in the reward system, encouraging the agent to balance accuracy with efficiency in its actions. "Update Policy" or "π" is the step where the reinforcement learning agent adjusts its policy based on the reward received from the system. This refines the agent's actions for future rule adjustments in the cycle, which again begins with "Observe State" or "S," completing the repeating cycle in reinforcement learning for SIEM rule optimization. This repeating cycle makes the SIEM system adaptive in nature, capable of adjusting to changing threats and network behavior while under supervision and control.

➢ *Dataset and Experimental Setup*

- *Dataset*

The research depends on the widely used benchmark datasets for intrusion detection. This is done in order to make the results reproducible and comparable with the previous research. The datasets used are NSL-KDD and CICIDS2017. The NSL-KDD dataset is an improved version of the original KDD Cup 1999 data. It is improved by the elimination of redundancy while providing a diverse collection of labeled network traffic data, categorized into normal and attack classes. The CICIDS2017 dataset is designed to replicate the characteristics of real-world network traffic, which is often considered realistic and modern. This means it contains not only malicious traffic but also benign traffic and various modern attacks like brute-force attacks, denial of service attacks, infiltration attacks, etc.

- *Simulation Environment*

A simulated SIEM environment is constructed to emulate operational conditions in which detection rules are parameterized and applied to streaming security events. Within this environment, dataset records are processed sequentially as if they were incoming network logs. Detection rules generate alerts based on configurable thresholds, correlation windows, and rule weights. The feedback from a SOC analyst is emulated by making use of the ground truth information already present in the data set. Once an alert is triggered, its accuracy is verified against the actual label present in the data set. This helps us calculate true positives, false positives, and false negatives. A reinforcement learning agent adjusts the rules based on the outcome and the rewards received. The entire process is done over multiple episodes, where each episode involves a complete pass over the data set or a selected subset of records.

➢ *Baseline Models*

To assess the effectiveness of the proposed optimization framework based on reinforcement learning, three different scenarios are defined for comparison. The first scenario is based on fixed configurations of the SIEM rules, where all thresholds and correlation values remain constant during the entire process. The second scenario is based on periodic tuning of thresholds for simulating the intervention process. The third scenario is based on a supervised machine learning classifier.

Table 2 Baseline Comparison Framework

Model Type	Adaptivity	Manual Effort	Expected FP Rate
Static SIEM Rules	None	High	High
Periodic Manual Tuning	Limited	Moderate	Moderate
Supervised ML (Offline)	Low	Moderate	Moderate
Proposed RL Optimization	Continuous	Low	Reduced

Table 2 shows the ways that each method can deal with the process of adaptation and the requirements for them to work. Static rules do not change on their own and require a lot of manual intervention, which results in many false alarms. Periodic tuning helps a little, but it depends on the schedule. Offline supervised learning can be used for

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

prediction, but it does not change its response to new threats unless it is retrained. The method based on reinforcement learning always optimizes, with minimal human intervention, striving for a constant decrease in false alarms.

➢ *Evaluation Metrics*

In the context of performance evaluation in intrusion detection systems, the basic classification and detection criteria are adopted. Accuracy is defined by the overall accuracy of the results. Precision is defined by the overall accuracy of the results. The false alert rate is defined by the overall accuracy of the results. The F1-score is defined by the overall accuracy of the results. The false negative rate is defined by the overall accuracy of the results. The alert volume is reduced, which is an indication of the improvement in the overall efficiency of the operation.

Table 3 Detection Performance Metrics

Metric	Formula	Objective
Accuracy	(TP + TN) / Total	Overall correctness
Precision	TP / (TP + FP)	Reduce false alerts
Recall	TP / (TP + FN)	Detect real threats
F1-Score	2(Precision × Recall) / (Precision + Recall)	Balanced measure
FPR	FP / (FP + TN)	Alert control
FNR	FN / (FN + TP)	Missed detection control

The quantitative measures we use to evaluate how good detection is and how good the system is at working can be seen in Table 3. Accuracy provides a general idea of how good it is, while precision and recall focus more on the trade-off between how good the alerts are and how good we are at detecting what we should be. The F1 score combines these. The percentage of false positives and false negatives provides a sense of the burden on the operation and security risks, respectively.

➢ *Experimental Procedure*

The experiment begins by creating the SIEM rule parameters, which define the baseline operation. The reinforcement learning agent is then given multiple episodes, during which it learns based on the sequential flow of the dataset's events. The agent observes the state of the system and takes actions by adjusting the rule parameters. It is rewarded based on the performance of the generated detections. After the reinforcement learning process, the best rule parameters are identified, which are those the reinforcement learning agent has learned. These are then input into the simulation environment. The performance is evaluated and compared with the baseline to determine the improvement. Statistical analysis is done to determine if the improvements in the detection rates or the reduction of false alarms are statistically significant. To ensure the results are not biased towards one particular data subset, cross-validation is done. This ensures the results are not based on one particular data traffic. The experimental design is critical in determining the contribution of the reinforcement learning approach towards the adaptive SIEM optimization.

➢ *Ethical Considerations*

The study uses publicly available datasets and simulated environments. No real organizational logs or personal data are processed. The reinforcement learning system is evaluated strictly for defensive security enhancement purposes.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

➢ *Experimental Setup Recap*

The NSL-KDD and CICIDS2017 data sets were utilized within the simulated environment, and the detection rules were set with specific threshold values to mimic the real environment [13]. The reinforcement learning agent was allowed to interact with the environment episodically, i.e., it observed the system, decided the way to alter the rules, and received rewards based on the performance of the detection system. Two different reinforcement learning strategies were compared, namely Q-Learning and Deep Q-Network. The Q-Learning strategy utilized a table to handle the states, but the DQN utilized a neural network to handle the complex state space. The models were trained for a large number of episodes until they converged. The performance of the model was compared with three different baselines.

➢ *Convergence Analysis*

To determine the system's rate of convergence, we observed the total reward accumulated during each successive episode of training. The total reward is an immediate measure of the policy's improvement, as it directly represents the optimal balance of detection and efficiency. A convergent system indicates that the agent has learned an effective strategy to adjust its own set of rules.

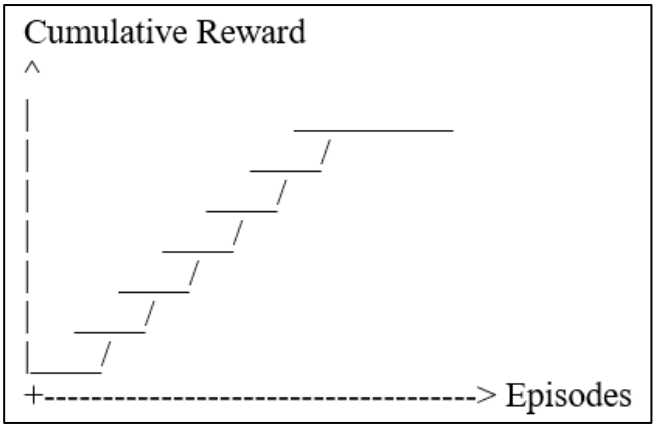

Fig 3 Cumulative Reward Convergence Trend

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

As can be seen in Figure 3, the rewards are always on the increase as the training process continues. This is an indication that the reinforcement learning agent has refined its policy and is performing almost consistently. The rapid stabilization of the Deep Q-Network is an indication of its advantage in dealing with state features.

➢ *Detection Performance Comparison*

After convergence, we calculated the performance metrics for the detection performance and compared them with the baseline systems to assess the effectiveness of adaptive optimization. We used accuracy, precision, recall, and F1-score to present a comprehensive view of the performance of the classifier.

Table 4 Detection Accuracy Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Static SIEM Rules	84.3	79.5	76.8	78.1
Manual Periodic Tuning	87.6	82.4	80.3	81.3
Supervised ML (Offline)	90.2	88.1	84.7	86.4
Q-Learning Optimization	92.8	90.5	89.1	89.8
DQN Optimization	95.4	93.8	92.6	93.2

From Table 4, it is clear that optimization through reinforcement learning outperforms all other static and manually configured scenarios in all performance parameters. Deep Q-Network has the highest accuracy, and the precision and recall are balanced, as indicated by the best F1 score among all the models.

➢ *False Positive and False Negative Reduction*

We looked deeper into how operations are affected by tracking false positives and false negatives. Reducing false positives can help reduce alert fatigue for the SOC team, and reducing false negatives can expand and strengthen our overall threat detection [10].

Table 5 Error Rate Comparison

Model	False Positive Rate (%)	False Negative Rate (%)
Static SIEM	14.2	18.7
Manual Tuning	11.5	15.4
Supervised ML	9.3	12.8
Q-Learning	7.6	9.1
DQN	5.2	6.4

Table 5, describes how false positives and false negatives steadily decrease as adaptivity increases. The reinforcement learning models have the lowest rates of false positives and false negatives, and the Deep Q-Network has the greatest improvement. Fewer false positives reduce analyst workload, and fewer false negatives improve defensive coverage.

➢ *Alert Volume and SOC Efficiency*

Alert volume reduction was evaluated by comparing the number of alerts generated per 10,000 processed events. This measure reflects the operational burden imposed on Security Operations Centers.

Table 6 Alert Volume Comparison

Model	Alerts per 10,000 Events	Reduction (%)
Static SIEM	1,820	—
Manual Tuning	1,540	15.4
Supervised ML	1,320	27.5
Q-Learning	1,080	40.7
DQN	890	51.1

From Table 6, it can be seen that the volume of alerts was considerably reduced by the reinforcement learning without compromising the detection quality. In fact, the Deep Q-Network results indicate that it was able to achieve more than 50% compared to the static rules, which can be considered a significant improvement in efficiency. This reduction in alerts can be considered a real gain for the SOC environment, which might be adversely affected by the large number of alerts.

➢ *Statistical Significance Testing*

In order to verify that the improvements provided by the reinforcement learning models are actually real, we decided to perform a paired t-test. We compared the RL-based configurations to the static settings of the SIEM rules over the same splits, using accuracy, F1 score, and false positive rate as metrics. For all metrics, the resulting p-values were less than 0.05, which means we can be at least 95% confident that the improvements are real. In other words, the improvements are not just the result of chance.

➢ *Robustness Against Concept Drift*

In order to determine the effectiveness of the system in adapting to changing threats, concept drift was simulated by adjusting the distribution of the data set from one training period to the next, simulating real-world changes in the

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

frequency of attacks, the nature of the traffic, and changes in behavioral patterns, among other factors, to which the reinforcement learning-based agent had to adjust by adapting the thresholds of the rules in real-time [11] [12].

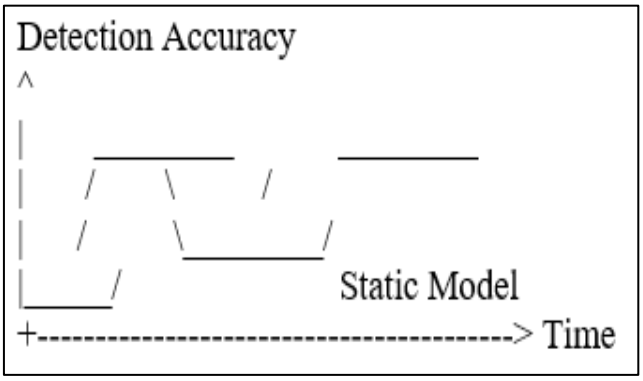

Fig 4 Adaptation Under Concept Drift

As can be seen in Figure 4, continuous learning helps in recovering the performance drop caused by the concept drift. The static setup remains static and gradually becomes ineffective, while the reinforcement learning framework recovers and maintains the accuracy of the detection by continually refining its policy.

➢ *Computational Overhead*

The computational overhead can be measured based on the time it takes for the training process and the overall cost incurred during the inference. In the tabular Q-Learning model, the overall latency was found to be minimal, as the decision-making process occurs based on simple lookups. However, the model required more episodes for the convergence process. In the case of the DQN model, the computational resources were found to be high, as the optimization process for the neural network was rigorous. However, the latency for the model, once deployed, remains minimal, making it suitable for real-time processing for the SIEM model. In the case of the bridge layer, which connects the RL module with the simulated SIEM engine, the overall processing time remains minimal.

V. DISCUSSION OF FINDINGS

As the experiments reveal, reinforcement learning is a viable way of ensuring that the rules of a SIEM remain optimized over time. Compared to a fixed setup of rules, the accuracy of the detection is much improved, and the RL models are able to achieve much higher precision and recall rates. The false positive rates are much reduced as well, which is a big help towards avoiding fatigue in Security Operation Centers that are operating the system. The adaptive nature of the system is quite strong as well, as it is able to adjust the rules quite well to remain stable through simulated concept drifts. An examination of the data reveals that the method is quite viable as the training costs remain low, and there is virtually no delay once the model is in place. Between the two models of RL that were presented, Deep Q-Networks was able to achieve better results than tabular Q-Learning on all fronts. The ability of DQN to deal with a more complex state is what gives it a much better chance of generalization to different kinds of data flows and rule configurations. The main way this is demonstrated is through a much quicker rate of convergence, much lower error rates, and a much better ability to deal with changes in data distribution.

➢ *Threats to Validity*

However, it is essential to point out some limitations despite the impressive findings. While benchmark data is great for reproducibility, it does not entirely represent the complexity and uncertainty of enterprise networks. Real-time traffic, on the hand, offers diversity in the form of systems, encryption, and organizational idiosyncrasies. In the paper, the feedback of the SOC was simulated using ground truth labels. In practice, humans have different opinions, and as such, there is an aspect of subjectivity in the determination of rewards. Additionally, the selection of the coefficients of the reward function influences the learning path and the optimization result. Different values of the weights could have produced different optimization results. The findings should be validated with real-time traffic and actual analyst feedback in order to increase external validity.

VI. CONCLUSION AND FUTURE WORK

➢ *Summary of the Study*

The research aims to address the challenges associated with static and manually tuned SIEM rules. The traditional rule-based approach is often found wanting as the attack mechanisms change, the users change, and the concepts change. The traditional approach is not scalable for large organizations. To solve the problems associated with the traditional approach, the research formulated the SIEM rule optimization problem as a Markov Decision Process. A reinforcement learning framework is developed. The RL model observes the environment, including the results of the rules, the feedback provided by the analysts, and the environment. It then optimizes the rules by adjusting the thresholds and correlation values. The paper developed an appropriate reward function that balances the need for accuracy and the need for efficiency. It is designed to penalize false positives, false negatives, and alert fatigue while rewarding true positives. The approach is implemented using the traditional Q-learning and the advanced Deep Q-Network approach. The results show that the proposed approach greatly improves the accuracy of the rules while reducing false positives, false negatives, and alert fatigue. The approach is effective even in the presence of concept drift.

➢ *Key Findings*

The research demonstrates the benefits of tuning the rules with reinforcement learning. The results show that the performance of the framework is enhanced. The models that employed reinforcement learning consistently outperformed the baseline models. The best accuracy and F1 score were obtained by the model employing the Deep Q-Network. The results demonstrate the benefits of employing adaptive policy learning in tricky scenarios. The takeaway is that the rule parameters are improved by continuous optimization. It is also evident that false positives are reduced. The framework is designed such that the reward function penalizes false positives while being sensitive to true positives.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

Consequently, as the reinforcement learning model improved its policy, the false positives reduced. The framework is beneficial in real-world scenarios. Too many false positives can lead to analyst fatigue and negatively impact the quality of the analysis. The framework reduces the false positives while maintaining the true positives. The framework is beneficial for improving the overall efficiency of the SOC. The framework is also beneficial in scenarios involving concept drift. The framework employing reinforcement learning can adapt the rules based on the changes in the network traffic. Consequently, the performance is improved. The framework employing static rules continued performing poorly. The framework is beneficial in real-world scenarios. The scenarios are dynamic, and the rules need to adapt. The framework employing reinforcement learning is beneficial. The framework is feasible. The results show that the model employing the Deep Q-Network required more computational resources during the training process. However, the deployment time is low. The optimization component has no negative impact on the SIEM workflow.

- *Theoretical Contributions*

This work contributes to the advancement of adaptive cybersecurity by employing a rule tuning problem for SIEM systems as an optimization problem for reinforcement learning. Rather than seeing rule tuning as a one-time engineering problem, this work presents a rule tuning problem as a sequential process where the agent learns to balance detection accuracy with operational efficiency. A specially designed reward function, which is applicable to real-world detection environments, facilitates this process. The reward function balances accuracy, false positives, and volume into a single objective function. This process aligns with the realities of the daily operation of the Security Operations Center. It goes beyond the accuracy-focused objectives often used in the context of the classification problem. Performance tests on standard intrusion detection datasets indicate improvements across multiple metrics. These results provide quantitative evidence to prove the effectiveness of the proposed adaptive rule learning to improve detection accuracy and operational efficiency. In addition, this work bridges the gap between the theory of reinforcement learning and the challenges faced by SIEM systems. Previous work focused on automated responses to threats. However, this paper employs reinforcement learning to the enhancement of rules within a real-world system.

- *Practical Implications*

From an operational standpoint, the proposed framework reduces the burden of constantly tuning the rules by hand. Conventional SIEM systems require constant human intervention to adjust the rules and filters, which is usually performed after receiving some reactive insights. The reinforcement learning agent optimizes this process constantly, allowing humans to concentrate on more strategic analysis. With improved accuracy of the alerts, resources for analysis can be utilized more effectively, resulting in faster and more efficient responses. The adaptive system can respond quickly to new threats, reducing the window of vulnerability before the detection system adjusts to the new behaviors. With constant optimization, the detection systems will improve over time, allowing them to withstand the test of time by gradually adjusting the parameters of the rules rather than making large-scale changes periodically. SOS can utilize reinforcement learning agents as optimization tools for fine-tuning the parameters of the rules while maintaining the essence of human oversight and governance.

- *Limitations of the Study*

Despite the encouraging results, some caveats are worth noting. The experiments were based on benchmark datasets rather than real, live traffic within an enterprise environment. While benchmarks are useful for reproducibility, they might not represent the full complexity of real-world networks. In the experiment, the feedback provided by the SOC was based on ground truth labels instead of real feedback. Real feedback would have come from the analysts, who would have provided some level of contextual understanding, some level of prioritization constraints, and some level of inconsistencies. The tuning of the reward weights is also an important factor in the overall behavior of the systems. The context in which the systems are deployed can require tuning the reward weights to balance the sensitivity of the detectors and the suppression of false positives. The paper also didn't discuss the potential for an attacker to manipulate the reward feedback. A sophisticated attacker could potentially manipulate the feedback mechanisms.

- *Recommendations for Future Research*

Future work involves checking how this framework works in real-world enterprise SIEM systems to see how it holds up under real-world conditions. Investigating multi-agent reinforcement learning could provide a way for distributed monitoring systems to learn together. For example, different agents could manage different groups of rules while still collaborating to ensure that detection is strong overall. Further work is necessary on the adversarial robustness of the RL agent. Checking how the RL agent holds up against attempts to manipulate or poison the reward signals could help increase trust in adaptive systems. Another promising area is combining this work with automatic response systems. Combining rule optimization with response optimization could provide a fully adaptive defense that can respond and adjust in near real-time. Developing explainable RL models could provide a way to increase trust in the adaptive defense by being able to provide clear reasons for changes to the rules. Policy transparency is particularly important for regulated or high-risk domains. Adding continual learning capabilities could provide a way to prevent catastrophic forgetting as threat patterns evolve over time. Continual learning could provide a way to learn new information without forgetting old information, thus maintaining effectiveness as the threat landscape changes.

- *Conclusion*

This research demonstrates the potential of reinforcement learning as an effective means of continually enhancing the SIEM rules. The potential of adaptive learning is such that security monitoring systems can improve from static configurations to sophisticated architectures. The results of the research demonstrate the potential of reinforcement learning in enhancing the accuracy of the

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar324

systems, reducing false positives, and improving the overall resilience against attacks. Though the results need to be validated in real-world scenarios, the framework developed in the research presents a good foundation for the development of adaptive security monitoring systems.

REFERENCES

[1]. G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial drift in intrusion detection systems," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2617–2631, 2021.

[2]. A. S. Aref, H. S. Hamza, and M. A. Hammad, "Reinforcement learning-based intrusion detection: A survey," *IEEE Access*, vol. 8, pp. 184379–184394, 2020.

[3]. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4]. Y. Lin, X. Liu, and J. Zhang, "Deep reinforcement learning for cybersecurity defense: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1016–1043, 2021

[5]. K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800-94, 2007.

[6]. W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," *IEEE Access*, vol. 5, pp. 21985–21990, 2017.

[7]. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[8]. V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9]. M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. Hoboken, NJ, USA: Wiley, 1994.

[10]. C. Gates and C. Taylor, "challenging the anomaly detection paradigm" in proc. ACM Workshop New Security Paradigms, 2006.

[11]. J. Gama et al., "A survey on concept drift adaptation" ACM computing surveys, vol. 46, no. 4, 2014.

[12]. S. Minku and X. Yao, "DNN ensembles for dealing with concept drift" IEEE Trans. Knowledge and Data Engineering, vol. 24, no. 4, pp. 619-633, 2012.

[13]. Ankit Thakkar and Ritika Lohiya, "A review of the advancement in intrusion detection datasets" precedia computer science 167 (2020) 636-645