# “Network Anomaly Detection for Intrusion Detection Systems Using Q-Learning and Deep Q-Learning: A Reinforcement Learning Approach”

2025 IEEE 9th International Conference on Software Engineering & Computer Systems (ICSECS)

**Adhitama Wichaksono, Parman Sukaro, Aulia Arif Wardana**

# Introduction

**Research Problem:**

- Modern cybersecurity systems face increasingly complex threats
- Signature-based IDS ineffective against zero-day attacks
- Anomaly-based IDS struggle with adaptation to network changes

**Research Objective:**

- Develop adaptive intrusion detection using Reinforcement Learning
- Compare Q-Learning and Deep Q-Learning (DQN) approaches
- Evaluate performance on zero-day attack detection

# Research Background

## Challenges in Network Security:

- Critical business processes depend on network systems
- Advanced attacks (DDoS, malware, zero-day) are increasing
- Traditional IDS limitations:
  - Rule-based systems ineffective against new attacks
  - High false positive rates hampering response times
  - Inability to adapt to changing network patterns

## Reinforcement Learning Potential:

- Enables agents to learn optimal detection strategies
- Adapts in real-time to new situations
- Reward-based learning suitable for cybersecurity applications

# Related Works

| Previous Studies: | Research Gaps: |
| --- | --- |
| • Modirrousta et al. (2022): DRL with CNN on UNSW-NB15 & BoT-IoT datasets - 99.17% accuracy<br>• Alavizadeh et al. (2022): Deep Q-Learning on NSL-KDD dataset - 78% accuracy<br>• Tellache et al. (2024): Multi-agent RL-DQN on CIC-IDS-2017 dataset - 99% accuracy | • Zero-day attack detection capabilities<br>• Handling imbalanced data<br>• Practical deployment considerations<br>• Real-world effectiveness evaluation |

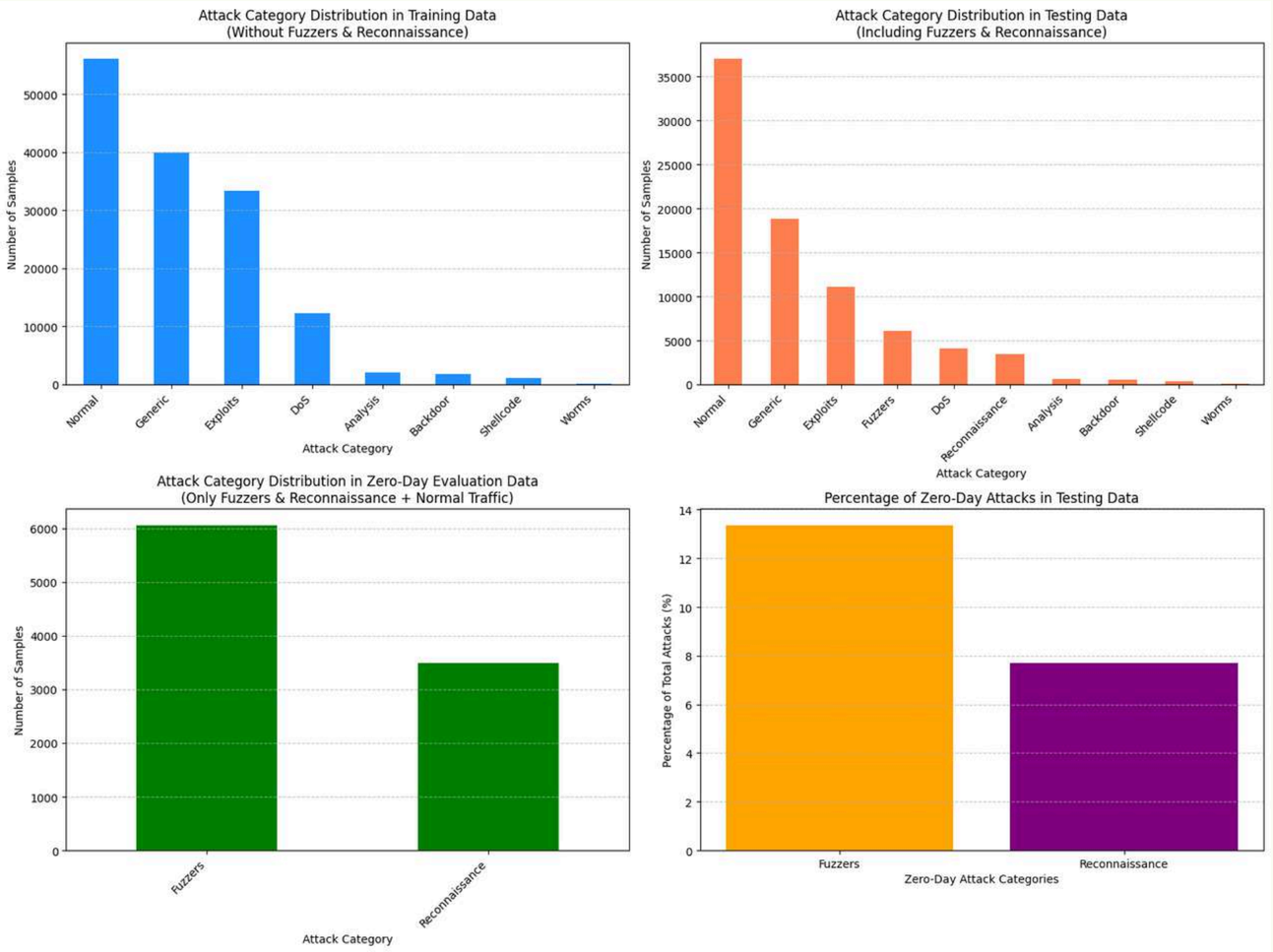
# Methodology Overview

| Dataset: | Zero-Day Simulation Approach: |
| --- | --- |
| • UNSW-NB15 dataset from Australian Centre for Cyber Security<br>• 175,341 training records, 82,332 testing records<br>• 45 features (IP addresses, port numbers, protocols, traffic metrics) | • "Fuzzers" and "Reconnaissance" attack types removed from training<br>• Retained in test set to evaluate zero-day detection<br>• Created specialized zero-day evaluation dataset (46,558 samples) |

# Data Preparation

**Dataset Distribution:**

- Training data: Excludes "Fuzzers" & "Reconnaissance"
- Test data: Includes all attack types
- Zero-day evaluation: Contains only "Fuzzers", "Reconnaissance", and normal traffic



# Data Preprocessing

**Preprocessing Pipeline:**

- Mean imputation for missing values
- Duplicate record removal
- Label encoding for categorical features
- SMOTE+Tomek for class imbalance (1:1 ratio)
- StandardScaler normalization
- Dimensionality reduction & feature selection

**Selected Features (20):**

| | | | |
|---|---|---|---|
| proto | sttl | djit | dmean |
| service | sload | swin | ct_srv_src |
| state | dload | stcpb | ct_state_ttl |
| dbytes | dinpkt | dtcpb | ct_src_ltm |
| rate | sjit | smean | is_sm_ips_ports |

# Model Development: Q-Learning

**Q-Learning Approach:**

- Model-free reinforcement learning algorithm
- Uses Q-table to store expected utility of actions

Updates Q-values based on rewards:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q^*(s',a') - Q(s,a))$$

**Implementation Details:**

- 30 states representing network traffic patterns
- Binary action space (2 actions): normal or anomalous
- Hyperparameter tuning with 5 different schemes

# Model Development: Deep Q-Learning

**DQN Approach:**

- Neural network replaces Q-table for Q-function approximation
- Experience replay buffer for stable learning
- Target network updated periodically

**Network Architecture:**

- Input layer (20 features)
- Two hidden layers with ReLU activation
- Output layer (2 actions)
- Trained to minimize loss:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim \mathrm{U}(D)} \left[ \left( r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]$$

# Experimental Setup

**Dataset Distribution:**

## Training

- 86,709 samples (50%),
- 50.0% normal,
- 50.0% attacks (Analysis, Backdoor, DoS, Exploits, Generic, Shellcode, Worms)

## General Testing

- 82,332 samples
- 56.0% normal
- 44.0% attacks (all categories)

## Zero-day Testing

- 46,558 samples,
- 62.5% normal,
- 37.5% attacks (only Fuzzers & Reconnaissance + Normal)

# Hyperparameter Tuning:

To optimize the performance of the Q-Learning and DQN models, hyperparameter tuning was conducted through five different schemes.

## Q-Learning Hyperparameter Tuning

| Scheme | Learning Rate (α) | Discount Factor (γ) | Initial Epsilon (ε) | Epsilon Decay | Minimum Epsilon |
|---|---|---|---|---|---|
| 1 | 0.01 | 0.8 | 1.0 | 0.8 | 0.05 |
| 2 | 0.1 | 0.95 | 1.0 | 0.95 | 0.2 |
| 3 | 0.03 | 0.99 | 0.8 | 0.99 | 0.3 |
| 4 | 0.5 | 0.9 | 0.7 | 0.9 | 0.1 |
| 5 | 0.2 | 0.7 | 0.9 | 0.85 | 0.01 |

## DQN Hyperparameter Tuning

| Scheme | Learning Rate | Epsilon Decay | Batch Size | Buffer Size |
|---|---|---|---|---|
| 1 | 0.0001 | 0.93 | 32 | 50000 |
| 2 | 0.0003 | 0.95 | 64 | 50000 |
| 3 | 0.0005 | 0.97 | 128 | 50000 |
| 4 | 0.001 | 0.98 | 256 | 50000 |
| 5 | 0.003 | 0.99 | 512 | 50000 |

# Results: Q-Learning Performance on Standard Test Data

## Key Observations:

- Scheme 1 achieved the best overall performance with highest accuracy and F1 score
- Low learning rate (0.01) with moderate discount factor (0.8) performed best
- All schemes achieved high precision (>97%), showing minimal false positives
- Recall varied more significantly (86-90%), indicating challenges in detecting all attacks

| Scheme | Accuracy | Precision | Recall | F1 Score | FPR | AUC |
|---|---|---|---|---|---|---|
| 1 | 0.9418 | 0.9903 | 0.9030 | 0.9447 | 0.0108 | 0.9825 |
| 2 | 0.9259 | 0.9950 | 0.8699 | 0.9282 | 0.0054 | 0.9690 |
| 3 | 0.9235 | 0.9943 | 0.8661 | 0.9257 | 0.0061 | 0.9727 |
| 4 | 0.9181 | 0.9760 | 0.8727 | 0.9215 | 0.0262 | 0.9577 |
| 5 | 0.9235 | 0.9788 | 0.8801 | 0.9268 | 0.0233 | 0.9666 |

# Results: DQN Performance on Standard Test Data

## Key Observations:

- All DQN schemes significantly outperformed Q-Learning
- Scheme 5 achieved the highest accuracy (99.70%) and F1 score (0.9972)
- Scheme 1 had exceptional recall (99.98%), detecting virtually all attacks
- All schemes demonstrated exceptional AUC values (>0.989)
- FPR remained low across all configurations (0.36-1.99%)

| Scheme | Accuracy | Precision | Recall | F1 Score | FPR | AUC |
|---|---|---|---|---|---|---|
| 1 | 0.9909 | 0.9840 | 0.9998 | 0.9918 | 0.0199 | 0.9999 |
| 2 | 0.9888 | 0.9941 | 0.9855 | 0.9898 | 0.0071 | 0.9898 |
| 3 | 0.9958 | 0.9971 | 0.9954 | 0.9962 | 0.0036 | 0.9973 |
| 4 | 0.9945 | 0.9935 | 0.9965 | 0.9950 | 0.0079 | 0.9991 |
| 5 | 0.9970 | 0.9969 | 0.9975 | 0.9972 | 0.0038 | 0.9994 |

# Comparison of Best Q-Learning and DQN Schemes: Zero-Day Attack Detection Results

DQN misclassified only 9 attack instances as normal, while Q-Learning misclassified 3,387 attacks.

## Performance on Unseen Attacks (Fuzzers & Reconnaissance):

| Model | Accuracy | Precision | Recall | F1 Score | FPR | AUC |
|---|---|---|---|---|---|---|
| Q-Learning | 0.9187 | 0.9393 | 0.6456 | 0.7653 | 0.0108 | 0.9472 |
| DQN | 0.9840 | 0.9283 | 0.9991 | 0.9624 | 0.0199 | 0.9994 |

## Confusion Matrices for Zero-Day Attacks:

| Model | TP | FP | TN | FN |
|---|---|---|---|---|
| Q-Learning | 6171 | 399 | 36601 | 3387 |
| DQN | 9549 | 737 | 36263 | 9 |

# Comparison with Traditional ML Models

Both reinforcement learning approaches outperformed traditional ML models, with DQN showing dramatically better zero-day detection than all alternatives.

## Performance on Standard Test Data:

| Metric | Q-Learning | DQN | Random Forest | SVM |
|---|---|---|---|---|
| Accuracy | 0.9418 | 0.9909 | 0.8978 | 0.8740 |
| Precision | 0.9903 | 0.9840 | 0.9462 | 0.9065 |
| Recall | 0.9030 | 0.9998 | 0.8634 | 0.8598 |
| F1 Score | 0.9447 | 0.9918 | 0.9029 | 0.8826 |
| FPR | 0.0108 | 0.0199 | 0.0601 | 0.1086 |
| AUC | 0.9825 | 0.9999 | 0.9667 | 0.9337 |

## Performance on Zero-Day Data:

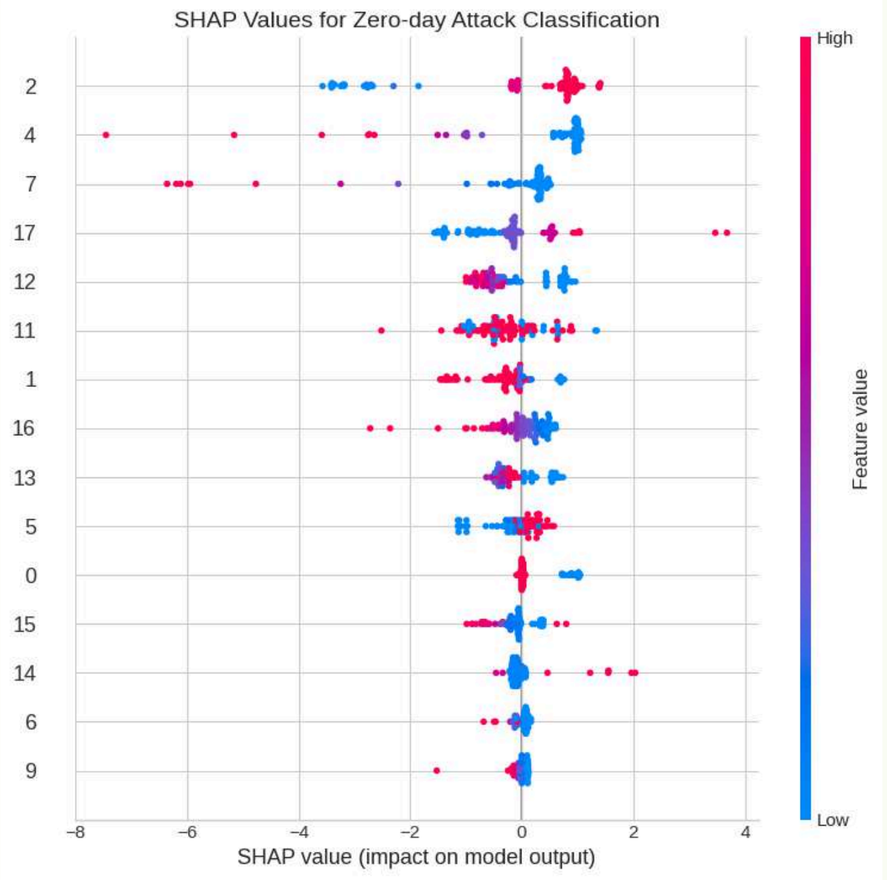
| Metric | Q-Learning | DQN | Random Forest | SVM |
|---|---|---|---|---|
| Accuracy | 0.9187 | 0.9840 | 0.8360 | 0.8063 |
| Precision | 0.9393 | 0.9283 | 0.6508 | 0.5314 |
| Recall | 0.6456 | 0.9991 | 0.4337 | 0.4768 |
| F1 Score | 0.7653 | 0.9624 | 0.5205 | 0.5026 |
| FPR | 0.0108 | 0.0199 | 0.0601 | 0.1086 |
| AUC | 0.9472 | 0.9994 | 0.8605 | 0.7588 |

# SHAP Values for Zero-day Classification

## Key Insights:

- Service feature (2) shows strongest impact for distinguishing zero-day attacks
- Connection metrics (ct_srv_src) clearly separate normal from malicious traffic
- Data bytes (dbytes) contribute significantly to detection capability
- The visualization confirms effectiveness of feature selection
- Model demonstrates adaptability to previously unseen attack vectors
- Features with highest SHAP values align with cybersecurity domain knowledge

**Detailed SHAP Analysis:**

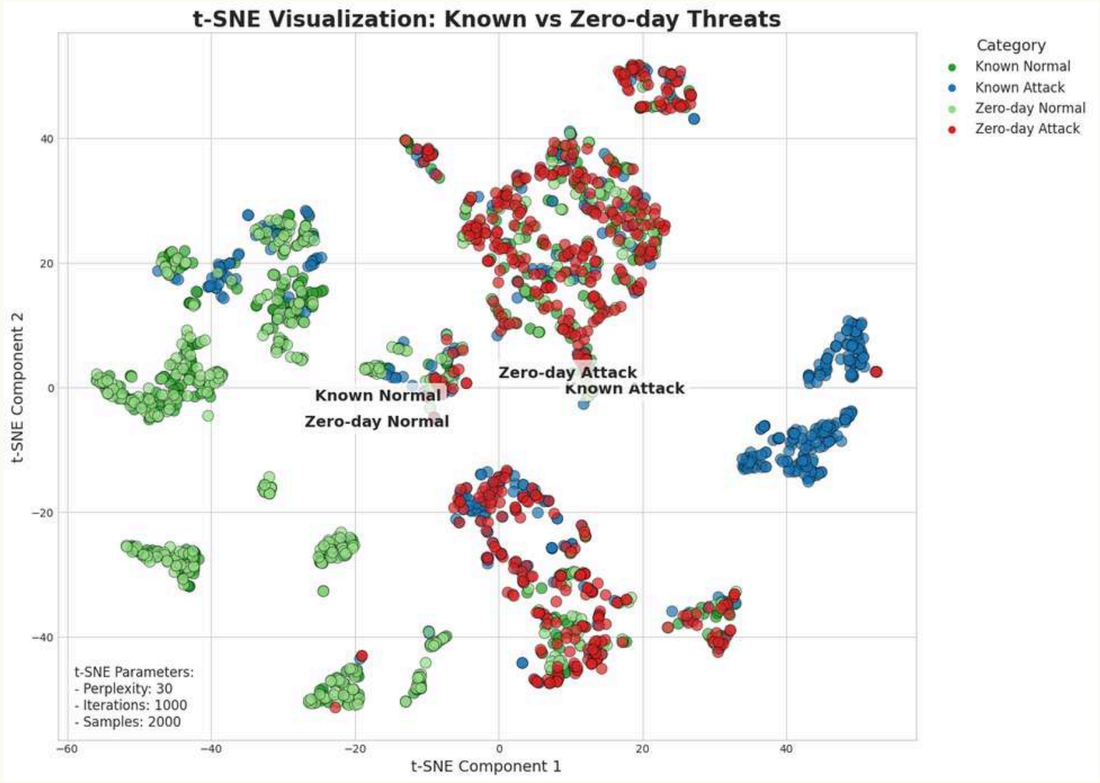


# t-SNE Visualization

## Visualization Insights:

- Four distinct clusters visible (Known Normal, Known Attack, Zero-day Normal, Zero-day Attack)
- DQN creates clear boundaries between normal and attack traffic
- Zero-day attack clusters positioned between known attacks and normal traffic
- Model learned generalized concept of "attackness" beyond specific signatures
- Known attacks form dense, well-defined clusters
- Zero-day attacks display more distributed patterns that partially overlap with known attack regions
- Effective transformation of feature space enhances attack-related patterns

**t-SNE Visualization of Traffic Patterns:**



# Computational Efficiency Comparison

## Performance Metrics:

| Model | Training Time (sec) | Memory Usage (MB) | Model Size (MB) | Inference Speed (samples/sec) |
|---|---|---|---|---|
| Q-Learning | 36.03 | 1,247.39 | 0.000458 | 997,931 |
| DQN | 4,852.5 | 2,572.1 | 0.03 | 3,623.4 |

## Efficiency Trade-offs:

- Q-Learning required only 0.74% of DQN's training time
- Q-Learning used 48.5% of DQN's memory
- Q-Learning model size dramatically smaller (0.000458 MB vs 0.03 MB)
- Q-Learning inference speed approximately 275 times faster
- These advantages must be weighed against DQN's superior detection capabilities
- Computational demands may limit DQN deployment in resource-constrained environments

| CONCLUSION | FUTURE WORK |
| --- | --- |
| • DQN proves to be more accurate in detecting network intrusions, achieving 99.09% accuracy compared to Q-Learning's 94.18%. DQN's advantage is most evident when facing new attacks (zero-day attacks) - its performance barely drops, while Q-Learning plummets to 64.56%.<br><br>• However, Q-Learning has its perks: it's lighter and faster. Training time is shorter, requires less memory, and the model size is smaller. | Next Research Directions:<br>• Develop hybrid models that combine DQN's detection accuracy with Q-Learning's computational efficiency<br><br>• Test and optimize real-time detection performance in live production environments |

# Thank you!

adhitamaw@student.telkomuniversity.ac.id
psukarno@telkomuniversity.ac.id
auliawardan@telkomuniversity.ac.id

Code Repository: https://github.com/adhitamaw/Anomaly-Detection-using-Reinforcement-Learning