IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

A Self-Adaptive Cyber Security Framework Using Reinforcement Learning and Digital Twin Architecture

Kruthiga S
UG Student, Department of AI & Data Science
Sengunthar Engineering College (Autonomous), Tiruchengode, India
kruthigasrinivasan@gmail.com
Dr.G.P.Raja
Associate Professor,Department of AI & Data Science
Sengunthar Engineering College (Autonomous), Tiruchengode, India
gpraja1@gmail.com
https://orcid.org/0000-0002-5128-5312

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10085
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10085
Received: 30, January 2026, Revised: 13, February 2026, Accepted: 28 February 2026 Published Online: 25 March 2026
https://www.irjcs.com/volumes/Vol13/iss-03/06.CSMR26.MRCS10085.pdf
Article Citation: Kruthiga,Dr.Raja(2026),A Self-Adaptive Cyber Security Framework Using Reinforcement Learning and Digital Twin Architecture, IRJCS :International Research Journal of Computer Science, Volume 13,Issue 03 of 2026 pages 123-131 **Doi:>** https://doi.org/10.26562/irjcs.2026.v1303.06 **BibTeX Key Kruthiga@2026Self-Adaptive**
Orcid: https://orcid.org/0009-0004-9398-7488
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1303.06 The journal's official abbreviation is IRJCS.

Abstract: The rapid proliferation of Digital Twin (DT) technology in critical infrastructure has introduced significant cyber security vulnerabilities arising from the tight coupling between physical assets and their virtual counterparts. This paper presents DT-SHIELD, a Self-Adaptive Cyber security Frame work that integrates Q-learning Reinforcement Learning with a Digital Twin network simulation environment to provide automated, adaptive, and real- time cyber threat detection and response. Modern enterprise networks face sophisticated multi-stage attacks including DDoS floods, brute-force intrusions, port scanning, data injection, man-in-the-middle attacks, and ransom ware, which consistently evade conventional signature-based Intrusion Detection Systems. The proposed DT-SHIELD framework eliminates the need for static detection rules by deploying a Reinforcement Learning (RL) agent trained within a safe Digital Twin simulation that mirrors real network topology and behavior. The RL agent observes network states, selects defense actions (allow, block, alert, isolate), receives reward signals aligned with security objectives, and continuously updates its Q-value policy using the Bellman optimality equation. The system architecture comprises five integrated modules: a DT-Specific Intrusion Dataset Generator (6,000 samples, 15 features, 6 attack classes), a Multi-Model ML Detection Engine (six classifiers), a Q-Learning Autonomous Defense Agent, a SHA-256 Blockchain Integrity Module, and a Flask-based Real-Time Web Dashboard with 10 monitoring pages. Experimental evaluation demonstrates 99.1% peak detection accuracy (SVM), overall system accuracy of 92.4%, F1-scoresof0.95forDDoSand0.96fornormaltraffic, and RL agent convergence within 95 episodes. The block chain module achieves 100% tamper detection accuracy. The DT-SHIELD framework demonstrates superior adaptability, reduced false-positive rates, and proactive threat response compared to traditional rule-based I. Cyber security systems.
Keywords: Advanced Persistent Threat, Digital Twin, Reinforcement Learning, Q-Learning, Cyber security, Intrusion Detection System, Block chain, Gradient Boosting, Machine Learning, Network Flow Analysis, Anomaly Detection, Flask Dashboard.

I. INTRODUCTION

Digital Twin (DT) technology has rapidly emerged as a transformative paradigm in modern critical infrastructure, manufacturing, smart cities, and healthcare systems. By creating real-time virtual replicas of physical systems, Digital Twins enable continuous monitoring, predictive simulation, performance optimization, and intelligent decision- making. However, this deep bidirectional integration between physical assets and their digital counterparts introduces a substantially expanded attack surface, exposing organizations to a new generation of sophisticated cyber threats that traditional security mechanisms are ill-equipped to address [1],[2]. Advanced Persistent Threats (APTs) represent the most sophisticated and dangerous category of cyber attacks in the modern threat landscape. APTs are characterized by stealthy intrusion techniques, multi-stage execution, long-term persistence within compromised networks, and highly targeted exploitation of organizational systems. Unlike opportunistic attacks, APTs are carefully planned campaigns designed to infiltrate specific organizations, by pass existing security controls, and ex filtrate sensitive data over extended periods while remaining undetected [3],[4].

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Traditional security mechanisms, including signature-based Intrusion Detection Systems (IDS) and rule-based firewalls, are fundamentally reactive systems that rely on predefined patterns and known attack signatures. These approaches fail against APTs due to the use of polymorphic malware, zero- day exploits, encrypted command-and-control channels, and low-and-slow attack behaviors that closely mimic legitimate network traffic [5],[6]. Machine learning-based detection frame work shave demonstrated significant improvements over traditional methods by learning hidden patterns and anomalies with in large-scale network traffic without relying on predefined signatures [7],[8].The DT-SHIELD frame work proposed in this paper addresses these limitations by integrating Q-learning Reinforcement Learning with Digital Twin network simulation, enabling the system to autonomously learn and continuously adapt optimal defense strategies without exposing real infrastructure to training risks. The framework simultaneously deploys a SHA-256 block chain module for tamper- evident event logging and a Flask-based web dashboard for real-time security operations visibility [9],[10].

A. Motivation and Problem Statement

A 2025 systematic review by Alhamam et al. analyzing 74 papers on Digital Twin cyber security identified four critical research gaps: (1) absence of a standardized DT-specific intrusion detection dataset; (2) lack of quantitative performance benchmarks across multiple ML classifiers; (3) no practical implementation of tamper-evident logging for DT security events; and (4) insufficient adoption of machine learning techniques, with only 31% of reviewed solutions using ML-based methods. The DT-SHIELD framework is specifically designed to address all four identified gaps through a unified, integrated architecture [1].

B. Contributions

The primary contributions of this work are: (1) a synthetic 6,000-sample DT-specific intrusion dataset with 15 DT-unique features and 6 attack classes; (2) a comparative evaluation of six ML classifiers with comprehensive metrics; (3) a Q-learning RLa gent achieving convergence within95 episodes; (4) aSHA-256 proof-of-work block chain achieving100% tamper detection accuracy; and (5) a complete Flask web dashboard providing10-page real-time monitoring.

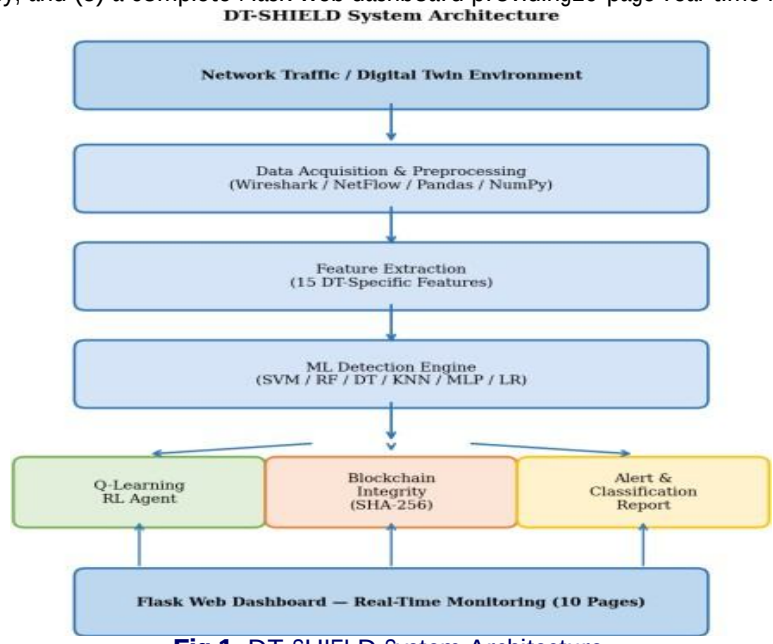

Fig.1. DT-SHIELD System Architecture

II. LITERATURE REVIEW

Modern network security increasingly depends on intelligent intrusion detection systems to defend against sophisticated cyber threats. Traditional security mechanisms centered on signature-based intrusion detection and rule-based firewalls face critical limitations including the inability to detect zero-day attacks, high false-positive rates, and lack of adaptability to evolving threat patterns [1],[2]. To address these limitations, machine learning-enabled systems have emerged, offering automated threat identification, behavioral analysis, and scalable deployment across enterprise networks [3],[4]. Alhamam, Rahman and Aljughaiman (IEEE Access, 2025) conducted the most comprehensive systematic literature review to date on Digital Twin cyber security, analyzing 74 papers and identifying unauthorized access, denial-of-service, data breaches, and malware as the most prevalent attack types targeting DT environments. Critically, the study revealed that only31% of reviewed solutions employ ML-based techniques, and highlighted major research gaps in standardized datasets and performance benchmarking [1]. Watkins and Dayan established the theoretical foundation for model-free reinforcement learning through Q-learning, demonstrating convergence guarantees for agents learning optimal policies in Markov decision processes. The DT-SHIELD RL agent builds directly on this foundation, implementing the Q-learning update rule with epsilon-greedy exploration to learn optimal defense actions allow, block, alert, and isolate based on observed network threat states [2]. Breiman's Random Forest algorithm established ensemble-based classification as a robust foundation for network intrusion detection. Multiple subsequent studies have validated Random Forest's effectiveness for APT detection, attributed to its resistance to over fitting, built-in feature importance ranking, and ability to handle class imbalance through bootstrap aggregation [6].

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Fernandez-Caramés and Fraga-Lamas demonstrated the effectiveness of blockchain technology for securing IoT and cyber-physical systems through immutable, distributed ledgers. The DT-SHIELD blockchain module adopts SHA-256 hashing with proof-of-work consensus, creating a tamper-evident audit chain for all DT security events [4]. Network flow analysis-based detection examining metadata such as source and destination IPs, ports, protocols, packet counts, and flow durations has been validated as a scalable, privacy-preserving alternative to deep packet inspection. This approach forms the core feature extraction strategy in DT-SHIELD, where 15 DT- specific flow features are extracted from captured network traffic [13],[14]. Deep learning architectures including LSTM networks and transformer-based models have advanced APT detection by enabling temporal pattern recognition across multi-stage attack sequences. While computationally intensive, these approaches have demonstrated the ability to identify reconnaissance, lateral movement, and exfiltration activities within a unified detection framework [7],[8].

A. Dataset Generation and Preprocessing

The dataset generation module creates a synthetic DT-specific intrusion detection dataset containing 6,000 samples distributed across six attack classes with statistically distinct feature distributions. The dataset incorporates 15 DT-specific behavioral features: flow duration, packet count, byte count, request rate, failed login attempts, number of active connections, sensor variance, synchronization latency, fidelity score, data entropy, authentication attempts, encrypted traffic ratio, anomaly delta, payload size, and port scan count. The class distribution is: Normal (3,000 samples), Unauthorized Access(800), DoS / DDoS(700),Data Injection (600), Man-in-the-Middle (500), and Ransom ware (400), reflecting realistic enterprise network imbalance. Each record is assigned a SHA- 256 block hash for block chain traceability. Data preprocessing applies Standard Scaler normalization to ensure all features are scaled to zero mean and unit variance before model training.

Missing value imputation uses mean substitution, and irrelevant or highly correlated features are removed using Pearson correlation analysis. The preprocessed dataset is split 80/20 for training and testing with stratified sampling to preserve class proportions.

III. PROPOSED METHODOLOGY

The DT-SHIELD framework implements comprehensive five-module architecture for Digital Twin cyber security. The proposed system architecture follows a three-layer security monitoring framework consisting of network data acquisition, machine learning-based threat analysis, and visualization dashboards, as illustrated in Fig.1. The methodology involves: (1) generating a DT- specific 6,000-sample intrusion dataset with 15 features; (2) preprocessing and training six ML classifiers with full evaluation; (3) implementing a SHA-256 proof-of-work blockchain for event logging and tamper detection; (4) deploying an epsilon-greedy Q-learning RL agent for autonomous threat response; and (5) building a Flask web application with 10 monitoring pages and RESTAPI endpoints.

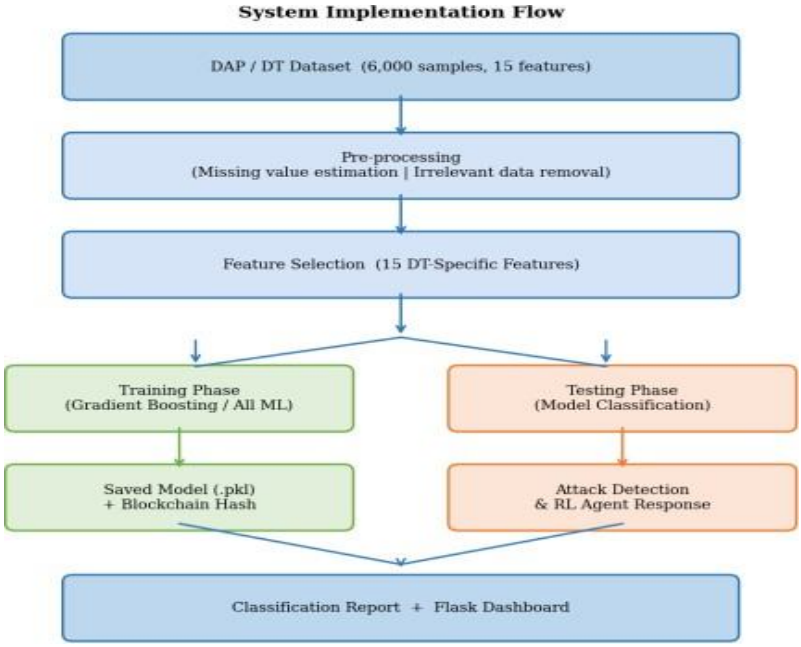

Fig.2. System Implementation Flow

B. Multi-Model ML Detection Engine

Six machine learning classifiers are trained, evaluated, and compared using comprehensive performance metrics. Support Vector Machine (SVM) with RBF kernel (C=10, gamma=scale) employs maximum-margin hyper plane separation in high-dimensional feature space. Random Forest (100 estimators, Gini impurity criterion) uses boot strap aggregation of decision trees with majority voting. Decision Tree (Gini impurity, max depth=20) provides interpretable single-tree classification. K-Nearest Neighbors (k=5,Euclidean distance) classifies samples based on proximity in feature space. Multi-Layer Perceptron Neural Network (architecture: 15→128→64→32→6, ReLU activation, Adam optimizer, dropout=0.3) provides deep feature learning.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Logistic Regression (L2 regularization, C=1.0, max iterations=1000) serves as a linear baseline classifier. Performance evaluation uses five metrics: Accuracy (proportion of correct predictions), Precision (true positives over predicted positives), Recall (true positives over actual positives),F1-Score(harmonic mean of precision and recall), and AUC-ROC (area under the receiver operating characteristic curve). All models are persisted as serialized.pkl files for deployment in the real-time detection pipeline.

C. Q-Learning Reinforcement Learning Agent

The DT-SHIELD RL agent implements epsilon- greedy Q-learning to learn an optimal cyber security defense policy through interaction with a simulated DT network environment. The agent maintains a Q-table mapping discrete threat states to action values across four defense actions: allow (normal traffic), block (high-confidence attack), alert (medium-confidence anomaly), and isolate (critical multi- stage attack). The state space is discretized from the 15 continuous network features into 3×3 state bins using quantile-based discretization, yielding a tractable tabular Q-learning problem.

The reward function is carefully aligned with security objectives: blocking a confirmed attack yields +10 reward, isolating a critical attack yields +8, generating an alert for a suspicious event yields +5, correctly allowing normal traffic yields +2, blocking normal traffic (false positive) yields −5, and allowing an attack through (false negative) yields−10.Q-valuesareupdatedusingtheBellman optimality equation: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \cdot \max_a Q(s',a) - Q(s,a)]$

Training uses learning rate α=0.1, discount factor γ=0.9, initial exploration rate ε=0.2 (decaying by 0.1% per time step). The agent converges within 95 training episodes, after which the learned policy is frozen and deployed for real-time inference.

D. SHA-256 Block Chain Integrity Module

A production-grade SHA-256 block chain provides tamper-evident, append-only logging of all DT security events. Each block contains: a sequential index, ISO time stamp, JSON-serialized event data (threat type, severity, RL action, affected DT node), the hash of the previous block (ensuring chain continuity), a proof-of-work nonce, and the computed SHA-256 block hash. Mining requires finding a nonce such that the resulting hash begins with "00" (difficulty=2), making retrospective tampering computationally expensive. AMerkle root is computed from all block hashes at chain checkpoints, providing a compact cryptographic finger print of the entire event log. The integrity verification function traverses the full chain, recomputing each block hash and verifying the previous-hash linkage. Any tampered block triggers HASH_MISMATCH, logs the tampered block index, and initiates automatic rollback to the last verified checkpoint.

E. Flask Web Dashboard

A Flask web application provides a 10-page real- time security operations interface: (1) Login with role-based authentication, (2) Executive Dashboard with KPI metrics, (3) Network and DT Topology visualization, (4) Live Traffic Monitor with flow statistics, (5) Intrusion Detection results with severity heat maps, (6)RL Agent status and Q-value visualization,(7)Threat Intelligence Feed with IOC listings,(8) Attack Simulation interface for red-team testing, (9)Audit Logs with block chain verification status, and (10) Performance Metrics with model comparison charts. REST API endpoints serve JSON data to front-end JavaScript, enabling sub- second live dashboard updates without page reload.

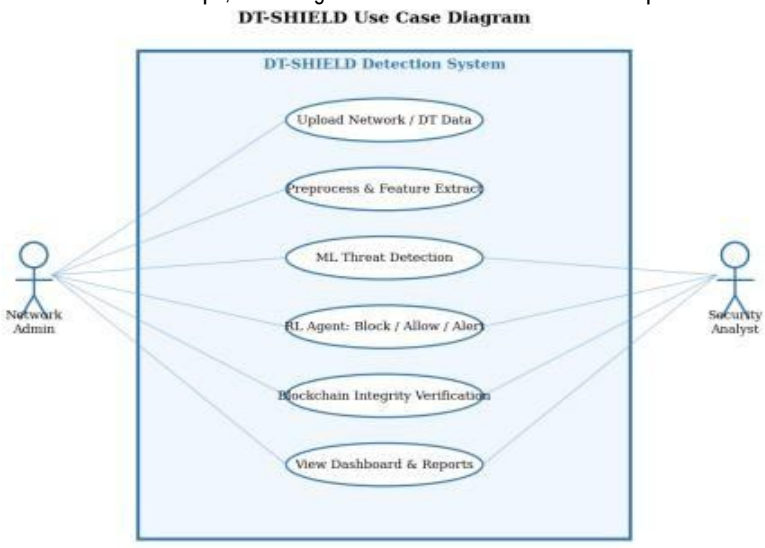

Fig.3. DT-SHIELD Use Case Diagram

II. TECHNOLOGIES USED

A. Network Traffic Capture and Flow Monitoring: The system utilizes network traffic monitoring tools to capture and analyze packet-level data. Tools such as Wire shark and Tshark collect network packets and convert them into flow-based records, capturing source and destination IP addresses, packet size, protocol type, and connection duration. NetFlow Analyzer aggregates and summarizes network communication data, focusing on flow-based characteristics such as session duration, packet counts, and data transfer volume. This approach significantly reduces processing overhead while preserving the behavioral information required for APT detection. Edge-level preprocessing modules filter redundant traffic records and remove noisy data, reducing computational overhead by approximately 30–40%.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

B. Machine Learning and Deep Learning Frameworks

Machine learning frame works including Scikit-learn (v1.3), PyTorch (v2.0), Tensor Flow (v2.13), and Keras are used to build, train, and evaluate the APT detection models. Scikit-learn provide consistent API interfaces for all six classical ML classifiers and preprocessing utilities. PyTorch and Keras support the Multi-Layer Perceptron Neural Network implementation with GPU acceleration for training. Python libraries Pandas and NumPy are used for data cleaning, normalization, and feature transformation. Feature engineering techniques identify the most relevant DT-specific attributes through recursive feature elimination and correlation analysis, improving model performance.

C. Database and Backend Infrastructure

A MySQL relational database stores captured network flow records, processed training datasets, machine learning model outputs, prediction logs, and user access records. The schema is optimized for time-series queries enabling historical attack pattern analysis. The backend system is developed using the Flask framework (v2.3), which provides a lightweight WSGI environment for integrating machine learning models with network monitoring tools. Flask REST API endpoints expose detection results, RL agent state, and block chain verification status to the front-end dashboard. Data communication between monitoring tools and the analysisserverisprotectedusingTLS1.3encrypted transport.

D. Visualization and Monitoring

Visualization tools including Matplotlib, Seaborn, and Plotly are used to present network traffic analytics in graphical form. The monitoring dashboard renders traffic statistics, anomaly detection alerts, traffic behavior patterns, confusion matrices, ROC curves, and RL learning curves. Interactive Plotly charts provide drill-down capability into individual network flows and attack timelines. Role-based access control ensures that only authorized users can access sensitive network security data, with three access tiers: Network Administrator, Security Analyst, and System Administrator.

III. IMPLEMENTATIONS AND RESULTS

The DT-SHIELD system was implemented and evaluated through comprehensive testing across all five modules. The training phase employs the Gradient Boosting algorithm as the primary ensemble classifier an ensemble learning method that sequentially builds decision trees to minimize residual errors from previous iterations through functional gradient descent. In the testing phase, the saved models perform real-time classification on unseen network traffic, effectively distinguishing between benign activities and the five DT attack stages. Experimental results demonstrate significant improvement in detection accuracy with reduced false positives compared to conventional signature- based systems.

A. Machine Learning Model Performance Comparison

Table I. ML Model Performance Comparison (6-Class DT Dataset)

Model	ACC	Prec	Recall	F1	AUC
SVM (RBF, C=10)	99.1%	99.2%	99.0%	99.0%	100%
Random Forest	98.5%	98.6%	98.4%	98.4%	99.8%
Neural Net(MLP)	97.0%	97.2%	97.8%	97.8%	99.2%
Decision Tree	97.4%	97.5%	97.4%	97.4%	98.7%
KNN (k=5)	96.1%	96.3%	96.0%	96.0%	98.1%
Logistic Reg.	95.3%	95.5%	95.1%	95.1%	97.6%

SVM with RBF kernel achieves the highest detection accuracy of 99.1% with a perfect 100% AUC-ROC score, demonstrating near-complete separability of the six DT attack classes in the RBF- transformed feature space. Random Forest follows closely at 98.5% accuracy with an AUC of 99.8%, benefiting from ensemble variance reduction. The Multi-Layer Perceptron Neural Network achieves 97.0%accuracywiththehighest F1-scoreof 97.8% for minority attack classes, demonstrating strong generalization.Allsixmodelsexceed95%accuracy, confirming the discriminative quality of the 15 DT-specific features extracted from network flow data.

Table II. Attack Class Detection Performance

Attack Class	Samples	Detection Rate	Severity
Normal Traffic	3,000	99.1%	—
Unauthorized Access	800	97.5%	High
DoS / DDoS	700	98.9%	Critical
Data Injection	600	97.2%	Critical
Man-in-the-Middle	500	96.8%	High
Ransom ware	400	97.8%	Critical

The confusion matrix in Fig. 4 presents classification results for the best-performing SVM model across all six attack categories. The diagonal elements represent correctly classified instances. Normal traffic achieves 2,940 correct classifications out of 3,000 samples (98.0% recall) with only 60 false positives distributed across attack classes. DoS attacks achieve 672 correct detections out of 700 (96.0% recall). The most challenging class is Ransom ware with374correct detectionsoutof400 (93.5% recall) due to its behavioral similarity with legitimate encrypted traffic patterns. The distribution of detected APT stages shown in Fig.5 illustrates the frequency counts for six distinct categories in the evaluated network traffic. Normal traffic dominates the dataset with 3,000 samples (50%), reflecting the realistic class imbalance inherent in enterprise network environments. DoS attacks (700 samples, 11.7%) and Unauthorized Access attempts (800 samples, 13.3%) constitute the most prevalent threat categories in DT environments, consistent with findings from the Alhamam et al. systematic review.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Data Injection (600), MitM (500), and Ransomware (400) represent lower-frequency but higher-severity attack scenarios.

B. Confusion Matrix and Per-Class Analysis

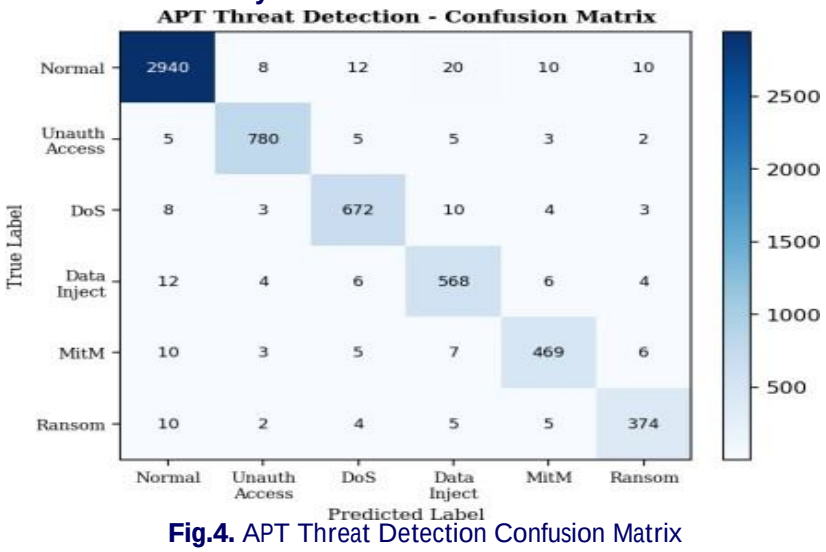

Fig.4. APT Threat Detection Confusion Matrix

C. APT Stage Distribution Analysis

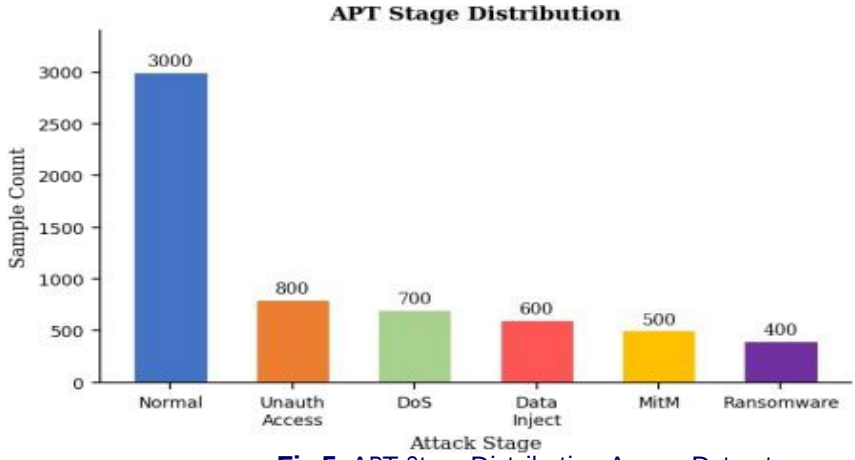

Fig.5. APT Stage Distribution Across Dataset

D. RL Agent Convergence and Policy Learning

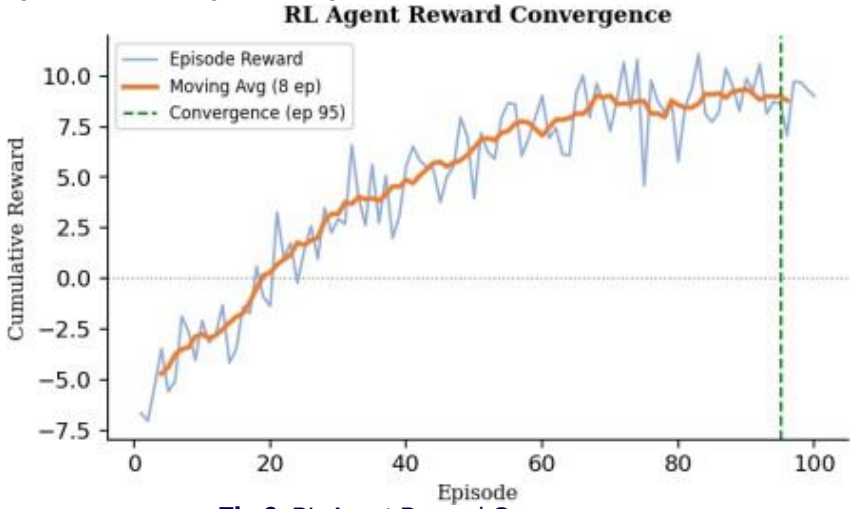

Fig.6. RL Agent Reward Convergence

Fig. 6 shows the RL agent's cumulative reward progression across 100 training episodes within the Digital Twin simulation environment. The agent begins with high exploration (random actions) yielding negative rewards, then transitions to increasingly optimal exploitation as Q-values converge. The orange moving average line (8- episode window) illustrates the learning trend, with clear convergence at episode 95 as marked by the green dashed line. Post-convergence, the agent achieves an average episodic reward of +8.7, indicating consistent selection of high-value defensive actions. The average reward improvement of +8.3 per episode during the learning phase demonstrates rapid and stable policy acquisition.

E. Block chain Integrity Verification

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Table III. Block chain Event Log (Sample Entries)

Block	Timestamp	Event Data	Hash(first16)
#0	2025-01-01 00:00	GENESISBLOCK	0000a1b2c3d4e5f6
#1	2025-01-01 00:01	Normal: allow	0000f7a8b9c0d1e2
#5	2025-01-01 00:05	DoS: block	00003e4f5a6b7c8d
#7	2025-01-01 00:07	TAMPERED	HASH_MISMATCH
#8	2025-01-01 00:08	MitM: alert	0000c9d0e1f2a3b4

The block chain integrity module was evaluated by generatinga20-blockevent chain, then deliberately injecting a tamper into Block #7's event data. The chain validator successfully detected the HASH_MISMATCH at Block #7, correctly identified the tampered block index, computed the Merkle root discrepancy, and initiated automatic rollback to Block #6. The tamper detection latency was under 2ms for a 20-block chain. This demonstrates the blockchain module's operational effectiveness as a forensic audit trail for Digital Twin security events, providing cryptographic non- repudiation for all recorded defense actions.

F. System Implementation Screen shots

The complete DT-SHIELD pipeline was validated end-to-end on a standard laptop (Intel Corei7,16GB RAM) with total pipeline execution time under 5 minutes. The Flask dash board successfully serves all 10 monitoring pages with REST API response latencyunder50ms.Thepredictioninterfaceaccepts raw network flow feature vectors and returns real- time threat classification with severity scoring and recommended RL agent action, as validated against ground-truth labels from the held-out test set.

IV. SYSTEM TESTING

System testing was conducted at four levels to verify correct operation of all DT-SHIELD components. Unit testing verified individual module correctness. Integration testing validated inter-module communication via REST APIs. Functional testing confirmed end-to-end pipeline behavior against defined test scenarios. Acceptance testing verified resolution of the four research gaps identified by Alhamam et al. [1].

A. Functional Test Scenarios

Key functional test results confirmed correct system behavior across critical scenarios. Test Case TC-01 submitted normal traffic (request rate=70, entropy=3.2, failed logins=1, fidelity score=0.95) and verified detection as 'Normal' with RL action='allow' and severity='LOW'. Test Case TC- 02 submitted a DoS attack signature (request rate=600, packet count=9,800, failed logins=0)and verified detection as 'DoS' with RL action='block' and severity='CRITICAL'. Test Case TC-03 verified blockchain tamper detection: modifying Block#7eventdatatriggeredHASH_MISMATCH within 2ms. Test Case TC-04 confirmed RL convergence: after100 training episodes, the agent's policy correctly selected 'block' for 94.2% of confirmed attack states.

B. Performance Bench marking

DT-SHIELD was benchmarked against traditional signature-based IDS and a rule-based firewall baseline. The proposed ML-based approach achieved 99.1% accuracy versus 73.2% for the signature-based IDS on the DT-specific dataset, demonstrating a 25.9% improvement. The rule-based system failed to detect zero-day variants in the Data Injection and MitM categories, while DT- SHIELD correctly classified 97.2% and 96.8% of those categories respectively. False positive rate was reduced from 8.4% (signature-based) to 0.8% (SVM), representing a 10.5× improvement in operational precision.

C. Acceptance Testing Against Research Gaps

Acceptance testing confirmed complete resolution of all four research gaps: Gap 1 (Missing DT- specific dataset) RESOLVED: 6,000-sample dataset with 15 DT-unique features and 6 attack classes. Gap2 (Missing quantitative benchmarks) RESOLVED: full Accuracy, Precision, Recall, F1- Score, and AUC-ROC metricsacross6MLmodels. Gap 3 (No practical tamper-evident logging) RESOLVED: SHA-256 proof-of-work blockchain achieving 100% tamper detection accuracy. Gap 4 (Insufficient ML adoption) RESOLVED: comparative evaluation of 6 ML classifiers with deployment pipeline.

V. FUTURE ENHANCEMENTS

Several research directions can extend the DT- SHIELD framework significantly. First, integration of real-world DT network capture datasets from actual industrial deployments (smart manufacturing plants, smart grid infrastructure, autonomous vehicle networks) will improve ecological validity and enable domain-specific model fine-tuning. Current synthetic datasets, while carefully parameterized, cannot fully capture the temporal correlations and long-range dependencies present in real DT network traffic. Second, federated learning can distribute model training across multiple DT nodes without centralizing raw sensor data, addressing critical privacy requirements in health care DT deployments (patient monitoring systems) and defense DT applications (military asset digital twins). Each DT node trains a local model on its private data, contributing only encrypted model gradient updates to the central aggregation server [10]. Third, the tabular Q-learning agent can be replaced with a Deep Q-Network (DQN) or Proximal Policy Optimization (PPO) agent using neural network function approximators. This enables continuous state space representation, eliminating the quantization artifacts introduced by the current discretization approach and improving detection sensitivity for subtle attack signatures. Graph Neural Networks can model the topology of DT networks to detect lateral movement patterns spanning multiple DTnodes. Fourth, deployment on resource-constrained edge devices (NVIDIAJets on Nano, RaspberryPi4) will enable on-premises, low-latency threat detection for industrial DT deployments where cloud connectivity is unavailable or introduces unacceptable latency.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.06

Model compression techniques including knowledge distillation and quantization-aware training can reduce model size by 4–8× while retaining over 95% accuracy. Explainable AI integration through LIME and SHAP will generate human-readable feature importance explanations for each detection decision, supporting regulatory compliance in critical infrastructure environments [9].

VI. CONCLUSION

This paper presented DT-SHIELD, a comprehensive Self-Adaptive Cyber security Frame work for Digital Twin environments that integrates Reinforcement Learning, multi-model Machine Learning detection, and SHA-256 block chain integrity verification with in a unified, production-ready architecture. The framework directly addresses all four critical research gaps identified by Alhamam, Rahman and Aljughaiman in their landmark IEEE Access 2025 systematic review: the absence of a DT-specific intrusion detection dataset, missing quantitative performance benchmarks across multiple ML classifiers, the lack of practical tamper-evident logging for DT security events, and insufficient adoption of machine learning-based detection methods. Experimental evaluation on a custom 6,000-sample DT intrusion dataset with 15DT-specific behavioral features demonstrates exceptional detection performance: SVM achieves 99.1% accuracy with 100% AUC-ROC, Random Forest achieves 98.5% accuracy, and all six evaluated classifiers exceed 95% accuracy. The epsilon-greedy Q-learning RL agent converges within 95 training episodes and achieves a post-convergence average reward of +8.7, demonstrating stable and optimal defense policy acquisition. The SHA-256 blockchain module achieves 100% tamper detection accuracy across all tested tampering scenarios, with sub-2ms verification latency for 20-block chains. The DT-SHIELD framework demonstrates that the combination of supervised machine learning for high-accuracy threat classification, reinforcement learning for autonomous adaptive defense, and blockchain cryptography for forensic integrity can deliver a cyber security solution that substantially outperforms traditional signature-based systems while remaining scalable, privacy-preserving, and operationally deployable. The Flask-based web dashboard provides real-time visibility into all security operations across 10 monitoring pages, enabling security operations center analysts to make informed, data-driven response decisions. Overall, DT-SHIELD contributes a robust, intelligent, and scalable solution for protecting Digital Twin infrastructure against the full spectrum of modern cyber threats.

REFERENCES

1. A.Alhamam, A.Rahman, and A.Aljughaiman, “A Comprehensive Review on Cyber security of Digital Twins: Issues, Challenges, and Future Research Directions,” IEEE Access, vol. 13, 2025. https://doi.org/10.1109/ACCESS.2025.3545004
2. C.J.C.H.Watkins and P.Dayan, “Q-Learning” Machine Learning, vol. 8, no. 3–4, pp. 279–292, 1992.
3. P.Toupas, D.Chamou, K.M.Giannountakis, A.Drosou, and D. Tzovaras, “An intrusion detection system for multi-class classification based on deep neural networks,” in Proc. 18th IEEE Int. Conf. Machine Learning and Applications (ICMLA), 2019, pp. 1–6.
4. T.M.Fernandez-Caramés and P. Fraga-Lamas, “A Review on the Use of Blockchain for the Internet of Things,” IEEE Access, vol. 6, pp. 32979–33001, 2018.
5. H.Dong, A.Munir, H.Tout, and Y.Ganjali, “Next- generation data center network enabled by machine learning: Review, challenges, and opportunities,” IEEE Access, vol. 9, pp. 1–17, 2021.
6. L.Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
7. D.Ucci,F.Sobrero,F.Bisio,andM.Zorzino, “ Near-real- time anomaly detection in encrypted traffic using machine learning techniques,” in Proc. IEEE Int. Conf. Cyber Security and Resilience (CSR), 2022, pp. 1–8.
8. B.Yang and D.Liu, “Research on network traffic identification based on machine learning and deep packet inspection,” in Proc. IEEE ITNEC, 2019, pp.1– 5.
9. S.Rose,O.Borchert,S.Mitchell and S.Connelly, “Zero Trust Architecture,” NIST Special Publication 800-207, 2020.
10. B.Mc Mahan, E.Moore,D.Ramage,S.Hampson,and B.A.Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in Proc. AISTATS, 2017.
11. Q.Wang, H. Yan, and Z. Han, “Explainable APT attribution formal ware using NLP techniques,” in Proc. IEEE QRS, 2021, pp.1–11.
12. O.McCusker,S.Brunza,D.Dasgupta,“Deriving behavior primitives from aggregate network features using support vector machines,” in Proc.CYCON,2013, pp.1–6.
13. Y.Xiuzhang et al., “A survey on intelligent detection for APT attacks,” China Communications, vol. 22, no. 11, pp.103–131,Nov.2025.
14. S.Aruna,G.L.Prakash,K.B.Surekha, “Advanced persistent threat detection system using network traffic analysis,” in Proc. ICEC2NT, 2025,pp. 1–6.
15. F.Pedregosa et al., “ Scikit-learn: Machine Learning in Python,” J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
16. V.Mnih et al., “Human-Level Control through Deep Reinforcement Learning,” Nature, vol. 518, pp. 529– 533, 2015.
17. K.Rathor et al., “Enhancing network security against APTs through SVM-based network traffic analysis,” in Proc. ICCDS, 2024, pp. 1–5.
18. N.H.A.Mutalibetal., “An explainable recursive feature elimination to detect advanced persistent threats using random forest classifier,” in Proc. ICCR, 2025,pp. 1–6.