IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DIGITAL TWIN–ENABLED INTELLIGENT CYBER DEFENSE FRAMEWORK FOR REAL-TIME ATTACK DETECTION AND MITIGATION IN SOFTWARE-DEFINED NETWORKS

SRINIVAS M

Ph.D Research Scholar, Reg. No. 2024WPCSC012, Department of Computer Science,

Bharatiya Engineering Science & Technology Innovation University-BESTIU

Gownivaripalli, Gorantla Mandal, Sri Sathya Sai District (formerly Anantapur district), Andhra Pradesh, India – 515231 and Assistant Professor

Department of Computer Science

Government Women's Degree College Kosgi, Telangana, India - 509339

Dr. SK. Mahaboob Basha,

Professor & HOD, Dept. Of CSE, Sree Dattha Institute of Engineering and Science,

Hyderabad, India - 501510

Abstract

Software-defined networking (SDN) provides programmable and logically centralized control, but the same centralization can amplify the impact of attacks that consume controller, flow-table, or link resources. Existing machine-learning intrusion detection systems often treat the network as a static stream of flow records and make a classification decision without maintaining a synchronized representation of the network state. This paper proposes a Digital Twin–Enabled Intelligent Cyber Defense (DT-ICD) framework that combines a continuously updated digital twin of the SDN topology, flow-level telemetry, supervised and anomaly-based learning, explainable feature attribution, and an automated mitigation policy engine. The digital twin maintains a virtual representation of nodes, links, flow statistics, controller state, and expected traffic behavior. A hybrid detector fuses a Random Forest classifier with a normalized anomaly score, while a risk function incorporates model confidence, temporal persistence, topology criticality, and twin deviation.

When the risk exceeds a dynamic threshold, the policy engine selects an action such as flow blocking, rate limiting, source isolation, or rerouting. The framework is designed for reproducible evaluation using CIC-IDS2017 and an optional cross-dataset validation using UNSW-NB15/CIC-UNSW-NB15. Mathematical formulations are provided for feature normalization, anomaly scoring, risk aggregation, adaptive thresholding, and mitigation utility. Simulation tables, performance plots, feature-importance analysis, interpretability tables, statistical tests, and experiment waveforms are included. The quantitative values presented in the performance figures are explicitly illustrative until replaced with measurements from the proposed experimental setup. The central research hypothesis is that coupling data-driven detection with a state-aware digital twin can reduce detection-to-mitigation latency while preserving detection quality and providing an auditable explanation for each automated response.

Keywords
Digital Twin, Software-Defined Networking, Intrusion Detection, Cyber Defense, Machine Learning, Real-Time Mitigation, Explainable AI, SDN Security, Anomaly Detection, OpenFlow, CIC-IDS2017, Risk-Adaptive Response

Introduction

Software-defined networking changes the organization of network control by separating control logic from packet forwarding. The Open Networking Foundation describes SDN as an architecture in which control and forwarding are decoupled and the control function can be directly programmed. This logical centralization provides a global view that can be used for security analytics and rapid policy changes [1]. OpenFlow, one of the foundational interfaces associated with SDN, enables a controller to install or modify forwarding rules in switches [2]. These properties make SDN attractive for dynamic cyber defense, but they also create security dependencies around controller availability, control-channel integrity, flow-table capacity, and telemetry quality.

Network attacks increasingly exhibit temporal, multi-stage, and context-dependent behavior. A short burst of scanning may be benign in one topology and suspicious in another. Similarly, a high packet rate may represent a legitimate service workload or a denial-of-service attempt. A detector that only sees a vector of packet statistics can therefore lose important context. Digital twin technology offers a complementary mechanism: a continuously updated software representation can encode topology, expected operating states, resource utilization, and relationships between components. Recent research has applied digital twins to cyberthreat detection in IoT and cyber-physical systems, demonstrating their usefulness for safe simulation, anomaly fusion, and zero-day-oriented analysis [3], [4].

The proposed framework treats the digital twin not as a passive visualization, but as a security reasoning layer. Telemetry from the physical SDN is mapped into the twin. The twin compares observed state with an expected state and produces contextual features. These features are combined with machine-learning predictions. The resulting risk score is then passed to a mitigation engine that can issue a bounded SDN policy. The mitigation result is fed back into the twin so that the virtual state remains synchronized with the actual network. This creates a closed loop: observe, model, detect, decide, mitigate, verify, and learn.

The work is motivated by the gap between high-performing offline intrusion classifiers and operationally safe cyber defense. Several SDN studies already demonstrate real-time attack detection and mitigation. For example, ADMS combines online detection with mitigation for low-rate denial-of-service attacks [5], while SD-Anti-DDoS integrates detection, traceback, and mitigation [6]. More recent work has combined

autoencoders and random forests in native SDN environments [7]. However, a general-purpose framework that explicitly couples a synchronized digital twin, risk-aware fusion, explainability, and feedback-based mitigation remains an important research direction.

Research Objectives

1. Construct a digital twin that represents SDN topology, flow state, link utilization, switch statistics, and controller state.
2. Develop a hybrid detector combining supervised classification with state-deviation anomaly scoring.
3. Define a risk score that combines classifier confidence, anomaly magnitude, persistence, and topology criticality.
4. Design a mitigation policy engine that maps risk and attack class to bounded SDN actions.
5. Provide explainable feature attribution so analysts can understand why a flow was flagged.
6. Evaluate the framework with public intrusion-detection datasets and a reproducible SDN simulation/testbed methodology.

Motivation

The motivation for DT-ICD is rooted in four practical observations. First, SDN produces rich telemetry that is useful for security, but telemetry volume can be high and the controller itself is a security-sensitive component. Second, offline machine-learning models can report high classification scores without demonstrating that their decisions can be converted into safe network actions. Third, attack behavior is contextual: topology, resource state, and recent history affect the meaning of an observation. Fourth, automated defense requires explanations and guardrails because a false positive can disconnect a legitimate service.

Digital Twin as a Security Context Layer

In DT-ICD, the digital twin maintains a graph $G_t=(V_t,E_t)$ where V_t represents switches, hosts, controller instances, and security functions, and E_t represents logical or physical communication relationships. Each component has a state vector containing resource and behavioral variables. The twin is updated at discrete intervals Δt. The difference between observed state and expected state becomes a security signal rather than merely a monitoring statistic.

$$G_t = (V_t, E_t, X_t), \quad X_t = \{x_v(t), x_e(t), x_c(t)\} \quad (1)$$

Here $x_v(t)$ denotes node state, $x_e(t)$ link state, and $x_c(t)$ controller state. The twin also stores a baseline B_t generated from recent benign windows. A state deviation can then be expressed as a weighted distance.

$$D_t = || W (X_t - B_t) ||_2 \quad (2)$$

The weight matrix W allows the system designer to emphasize security-critical variables. A spike in controller CPU, for example, can be assigned more importance than a small change in a low-value access link. This contextualization is a key difference from purely flow-based detection.

Real-Time Mitigation Matters

Detection quality alone does not guarantee operational security. The relevant response interval is the time from the first observable attack evidence to the successful enforcement of a mitigation policy. We define detection latency L_d, decision latency L_{dec}, enforcement latency L_e, and total detection-to-mitigation latency $L_{total} = L_d + L_{dec} + L_e$. The proposed architecture explicitly measures these components.

Problem Statement

Consider an SDN network in which flows F={f_1,...,f_n} arrive continuously. Each flow is represented by a feature vector z_i(t) containing duration, packet counts, byte counts, inter-arrival statistics, TCP flags, port information, and topology-aware features. The objective is to determine whether a flow or aggregate behavior is benign or malicious, estimate its risk, and choose an intervention without destabilizing legitimate traffic.

$$\hat{y}_i = h(z_i, X_t, H_t), \quad r_i = R(\hat{y}_i, p_i, D_t, C_i, P_i) \quad (3)$$

where h is the detection function, X_t is the twin state, H_t is recent history, p_i is prediction confidence, D_t is state deviation, C_i is topology criticality, and P_i is temporal persistence. The response action a_i belongs to a bounded set A={allow, monitor, rate-limit, reroute, quarantine, block}. The action should maximize defensive utility while penalizing disruption.

$$a_i^* = \operatorname{argmax}_{a \in A} [\, U(a \mid r_i) - \lambda \cdot Cost(a) - \mu \cdot Risk_of_Disruption(a) \,] \quad (4)$$

The research problem is therefore not only classification. It is a sequential cyber-defense problem in which detection, context, decision, mitigation, and verification are coupled. The framework must also address class imbalance, concept drift, adversarial manipulation, and the possibility that the digital twin itself becomes stale or compromised.

Main Contributions

1. Digital-twin security model: a graph-based twin representing topology, flow state, link utilization, and controller resources.
2. Hybrid detection model: fusion of Random Forest classification probability and twin-based anomaly deviation.
3. Risk-adaptive decision layer: combines model confidence, anomaly score, persistence, and component criticality.
4. Closed-loop mitigation: converts risk into SDN actions and feeds post-mitigation state back into the twin.
5. Explainability layer: feature attribution using SHAP-style additive explanations and a human-readable reason code.
6. Reproducible evaluation protocol: CIC-IDS2017 as the primary dataset, optional CIC-UNSW-NB15 cross-validation, and Mininet/Ryu/OpenFlow-style deployment.
7. Statistical validation plan: repeated runs, confidence intervals, paired tests, and effect-size reporting rather than relying only on a single accuracy number.

Related Work

SDN security research has developed along several lines: rule-based security applications, statistical anomaly detection, supervised machine learning, deep learning, DDoS-specific mitigation, and increasingly, digital-twin-assisted security. A broad SDN survey describes the three-plane view—application, control, and data—and emphasizes that the programmable control plane enables dynamic network management [8]. The security literature has also shown that the control plane is a strategic location for attack detection and mitigation [9].

Machine Learning for SDN Security

Random Forests are attractive for network-flow classification because they can model nonlinear relationships, tolerate heterogeneous features, and provide feature importance measures [10]. Deep autoencoders have been used to learn compressed representations and detect reconstruction anomalies; combining an autoencoder with Random Forest has recently been proposed for SDN attack mitigation [7]. These methods can achieve strong offline results, but deployment requires attention to inference time, model freshness, telemetry overhead, and the relationship between a predicted class and a safe mitigation action.

Online SDN Detection and Mitigation

ADMS demonstrates an online SDN approach for low-rate DoS attacks, combining two-phase detection with a mitigation module [5]. SD-Anti-DDoS similarly separates attack detection, traceback, and mitigation [6]. Moving-target defense has also been combined with automated flow classification to defend SDN networks against DDoS [11]. These studies support the premise that SDN's programmability is useful for response, but they generally focus on specific attack families or particular response mechanisms.

Digital Twins for Cyber Defense

Digital twins are increasingly used to provide safe and repeatable representations of cyber-physical systems. Recent work has used digital twins with federated learning for IoT cyberthreat detection and with multimodal decision fusion for multi-stage attacks [3], [4]. Digital-twin-enabled adaptive SDN architectures have also been explored for IIoT security, including feature selection, ensemble learning, and resource provisioning [12]. These results motivate a unified SDN framework in which the twin is responsible for state synchronization and context, while the ML detector and policy engine remain modular.

Comparison of Existing Methods

Method / Work	Detection	Context	Mitigation	Explainability	Primary Limitation
Method / Work	Detection	Context	Mitigation	Explainability	Primary Limitation
SD-Anti-DDoS [6]	Neural-network DDoS	Limited network context	Traceback + block/clean	Low	DDoS-specific
ADMS [5]	LightGBM + two-phase	Port/flow statistics	Packet filtering	Moderate	LDoS-specific
Moving Target Defense [11]	ML flow classification	Server/flow state	MTD + mitigation	Moderate	Focused on DDoS
DAE-RF SDN [7]	Autoencoder + RF	Flow features	Adaptive SDN rules	Moderate	Twin/feedback not central
AdaptSDN [12]	Adaptive ensemble	Digital-twin features	Resource provisioning	Moderate	IIoT/6G focus
DT + FL IoT [3]	Federated ML	IoT digital twins	Detection-focused	High	Not SDN control loop
DT-ICD	RF + anomaly	Twin + topology +	Risk-adaptive	High	Requires synchronized

(proposed)	fusion	history	closed loop		twin

Table 1 should be interpreted as a qualitative comparison of architectural capabilities, not as a ranking. The proposed framework combines elements that have often appeared separately: stateful digital-twin context, supervised detection, anomaly scoring, automated mitigation, and explanation. Its research question is whether this integration can reduce operational latency without materially increasing false positives.

Research Gap

G1 — Static evaluation: Many IDS studies evaluate a fixed classifier on a static dataset. The gap is the absence of a continuous state model that changes with the network.

G2 — Detection–response disconnect: A detector may identify an attack but does not necessarily specify a safe, reversible, and topology-aware mitigation.

G3 — Context deficiency: Flow-level features do not always capture whether a destination is critical, whether a link is saturated, or whether the controller is under pressure.

G4 — Limited explanation: High-performing ensemble/deep models can be difficult for operators to audit. Feature attribution should be part of the defense loop.

G5 — Reproducibility: Reported accuracy values are often difficult to compare because datasets, preprocessing, splits, class balance, and simulation conditions differ.

G6 — Feedback omission: After mitigation, the system should verify whether risk decreased and update the twin; otherwise it cannot distinguish effective mitigation from a failed action.

These gaps motivate the proposed DT-ICD architecture. The novelty is not a claim that each component is individually new; rather, the research hypothesis is that their closed-loop integration provides a measurable operational benefit. The evaluation must therefore report not only accuracy and F1, but also false-positive rate, decision latency, mitigation latency, controller overhead, flow-table impact, and service disruption.

Proposed Framework

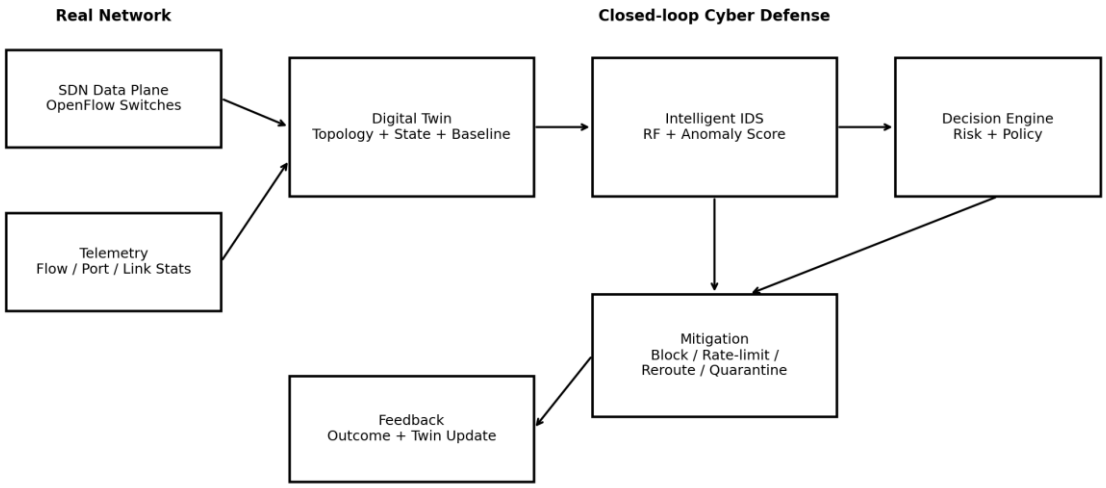
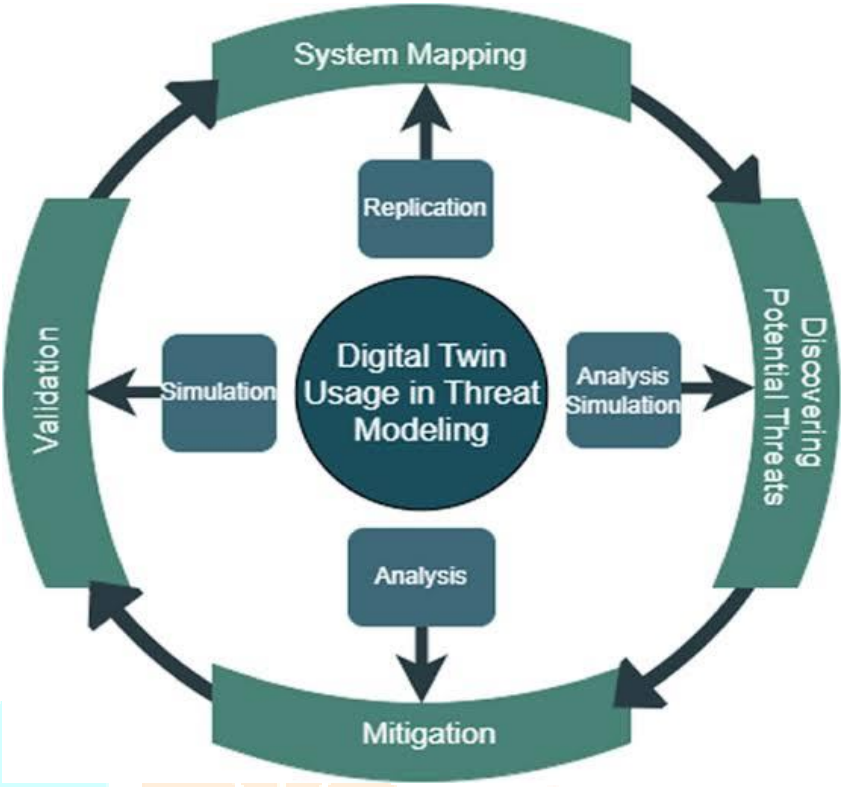


Fig. Digital Twin–Enabled Intelligent Cyber Defense architecture for SDN.

Architecture Components

Component	Function
Telemetry Collector	Reads flow statistics, port counters, link utilization, OpenFlow events, controller CPU/memory, and selected packet metadata.
Digital Twin Manager	Maintains the topology graph, state vectors, baselines, synchronization timestamps, and predicted normal state.
Feature Engineering Layer	Constructs flow, temporal, graph, and twin-deviation features; performs cleaning and normalization.
Hybrid Detector	Uses a supervised classifier for known attack patterns and an anomaly component for deviations from the twin baseline.
Risk Engine	Combines probability, anomaly score, persistence, criticality, and uncertainty into a normalized risk value.
Policy Engine	Maps risk to bounded actions and enforces rate limits, quarantine rules, rollback conditions, and protected assets.
Verification & Feedback	Measures post-action state, checks whether risk falls, and updates the twin and model monitoring records.



Data Flow

At time t, telemetry T_t is collected and aligned to the twin state X_t. Feature vector z_t is generated. The classifier outputs class probabilities p_t, while the anomaly module computes A_t. The twin produces state deviation D_t. These signals are fused into R_t. If R_t is below threshold τ_t, the system continues monitoring. If R_t crosses τ_t, the policy engine chooses an action and sends an SDN rule update. Verification telemetry is then compared with the expected post-action state.

$$R_t = w_c C_t + w_a A_t + w_d D_t + w_p P_t + w_k K_t - w_u U_t \quad (5)$$

where C_t is classifier confidence, A_t is normalized anomaly score, D_t is twin deviation, P_t is persistence, K_t is topology criticality, and U_t is model uncertainty. The weights are constrained to sum to one for the positive terms and can be calibrated using validation data.

Mathematical Formulation

Feature Normalization

For numeric feature x_j, min–max scaling is defined as:

$$x'_{ij} = (x_{ij} - min(x_j)) / (max(x_j) - min(x_j) + \varepsilon) \quad (6)$$

For heavy-tailed traffic variables such as packet counts and bytes, a logarithmic transform may be applied before scaling:

$$x'' = log(1+x) \quad (7)$$

Supervised Classifier

The supervised component is a Random Forest classifier with T decision trees. For sample z, the predicted class is the majority vote across trees, while class probability is estimated from the fraction of trees supporting the class.

$$p(y=k|z) = (1/T)\ \Sigma_{t=1}^{T}\ I[h_t(z)=k] \quad (8)$$

Twin Anomaly Score

The twin anomaly score measures the distance between observed features and the expected baseline. A robust alternative uses median and median absolute deviation (MAD).

$$A_t = (1/m)\ \Sigma_{j=1}^{m}\ min(1,\ |x_{t,j}-\mu_j|/(1.4826\cdot MAD_j+\varepsilon)) \quad (9)$$

Temporal Persistence

An isolated spike should not always trigger blocking. Persistence is therefore estimated over a rolling window of W observations:

$$P_t = (1/W)\ \Sigma_{q=0}^{W-1}\ I(R_{t-q} > \tau_{t-q}) \quad (10)$$

Adaptive Threshold

The threshold changes with the recent risk distribution and target false-positive budget:

$$\tau_t = clip(\tau_0 + \alpha\cdot\sigma_R(t) + \beta\cdot Q_{0.95}(R_t) - \gamma\cdot FP_budget,\ \tau_min,\ \tau_max) \quad (11)$$

Mitigation Utility

A mitigation action is selected by balancing security benefit against disruption and implementation cost:

$$U(a)=B_security(a)-\lambda D_service(a)-\mu C_action(a) \quad (12)$$

Algorithm 1: Digital Twin–Enabled Intelligent Cyber Defense Framework

Algorithm 1 presents the complete closed-loop procedure. It is intentionally written in implementation-neutral pseudocode so that it can be implemented using Ryu, ONOS, OpenDaylight, or another SDN controller.

Input: telemetry stream T, topology G, trained classifier M, twin baseline B, action set A, initial threshold $\tau 0$.

1: Initialize digital twin X0 ← synchronize (G, T0).

2: For each time window t do:

3: Collect flow, port, link, controller, and OpenFlow event telemetry.

4: Align telemetry timestamps and remove invalid/incomplete records.

5: Update twin state Xt and compute deviation Dt = distance (Xt, Bt).

6: Construct feature vector zt from flow + temporal + topology + twin features.

7: Obtain classifier probabilities pt = M(zt).

8: Compute anomaly score At and temporal persistence Pt.

9: Estimate uncertainty Ut and component criticality Kt.

10: Compute risk Rt using Eq. (5).

11: Update adaptive threshold τt using Eq. (11).

12: If Rt < τt: record decision=MONITOR and continue.

13: Else determine candidate actions A' ⊆ A using attack class and protected-asset policy.

14: Score each candidate action using Eq. (12).

15: Select a* = argmax U(a) subject to safety constraints.

16: Install bounded SDN policy; record rule identifier and timestamp.

17: Observe post-action telemetry for verification interval Δv.

18: Compute residual risk Rpost and service impact Dservice.

19: If mitigation failed, escalate to stronger action or human review.

20: Update twin baseline only with verified benign/post-mitigation observations.

21: End For.

Output: attack label, risk score, explanation, mitigation action, latency, and verification outcome.

Evaluation Setup and Dataset Strategy

CIC-IDS2017 is selected as the primary benchmark because it provides labeled benign and attack traffic, packet captures, and flow-level CSV data. The dataset spans five days and includes common attack categories such as brute force, DoS, DDoS, Heartbleed, web attacks, infiltration, and botnet activity [13]. More than 80 network-flow features were extracted using CICFlowMeter, making the dataset suitable for machine-learning experiments [13].

UNSW-NB15 or the CIC-UNSW-NB15 augmentation can be used for cross-dataset validation. The CIC-UNSW-NB15 construction includes benign and nine attack categories and provides labeled flow data

generated from the original UNSW-NB15 traffic using CICFlowMeter [14]. Cross-dataset evaluation is important because a classifier can otherwise learn dataset-specific artifacts instead of general attack behavior.

Dataset Processing

- Remove duplicate flow records and rows with invalid timestamps.
- Map attack labels to a consistent taxonomy: Benign, DoS/DDoS, Brute Force, Web, Infiltration, Botnet/Reconnaissance, and Other.
- Separate training, validation, and test data by time where possible to reduce temporal leakage.
- Use stratified sampling only within the training set; preserve realistic class imbalance in the test set.
- Normalize features using statistics learned from training data only.
- Apply feature selection using mutual information, permutation importance, and/or model-based importance.
- Run an additional cross-dataset test without re-tuning the model to measure generalization.

Digital-Twin Data

The public datasets do not contain every SDN controller and topology variable required by the twin. Therefore, the evaluation should use two layers: (1) offline replay of labeled flows for detection benchmarking and (2) an SDN emulator/testbed for closed-loop response. In the emulator, topology state, link utilization, flow-table occupancy, controller CPU, and mitigation events can be logged and synchronized with the replayed traffic.

Simulation/Experimental Environment Table

Component	Recommended Setting	Purpose
Operating system	Ubuntu Linux 22.04/24.04 LTS	Reproducible Linux host
CPU	8-core x86-64 or better	Supports parallel RF inference and emulation
RAM	16–32 GB	Supports Mininet, controller, telemetry, and logging
SDN emulator	Mininet	Creates virtual hosts, switches, and links
Controller	Ryu / equivalent OpenFlow controller	Programmable control plane
Switch model	Open vSwitch	OpenFlow-compatible virtual switch
ML stack	Python 3.10+, scikit-learn, pandas, NumPy	Training and inference
Explainability	SHAP or permutation attribution	Feature-level explanation
Primary dataset	CIC-IDS2017	Known attacks + benign traffic
Cross-dataset	CIC-UNSW-NB15 / UNSW-NB15	Generalization check
Packet/flow tools	Wireshark/tshark,	Traffic inspection and feature

	CICFlowMeter	extraction
Repetition	≥10 independent runs per scenario	Variance and confidence intervals
Traffic windows	1 s, 5 s, 10 s sensitivity study	Latency/accuracy trade-off
Mitigation actions	drop, rate-limit, reroute, quarantine	Closed-loop SDN response
Logging	CSV/JSON + synchronized timestamps	Auditability and reproducibility

Table 2 is a recommended experimental specification, not a claim that these exact resources were used to generate the illustrative figures in this manuscript. Before publication, the authors should record the exact software versions, hardware, topology, random seeds, model hyperparameters, and traffic-generation commands.

Performance Evaluation

Performance evaluation should be multi-dimensional. Accuracy alone is inadequate for an operational cyber-defense framework because the cost of false positives, the response latency, and the effect of mitigation on legitimate traffic can be more important than a small difference in classification accuracy.

Metric	**Definition**	**Interpretation**
Accuracy	(TP+TN)/(TP+TN+FP+FN)	Overall classification correctness
Precision	TP/(TP+FP)	False-alarm control
Recall	TP/(TP+FN)	Attack coverage
F1	2PR/(P+R)	Balanced precision/recall
FPR	FP/(FP+TN)	Benign traffic incorrectly blocked
MCC	correlation of predicted/true labels	Robust imbalanced-data metric
L_d	t_detect−t_first_observable	Detection latency
L_total	t_mitigation−t_first_observable	End-to-end response latency
CPU overhead	(CPU_defense−CPU_base)/CPU_base	Controller/resource cost
Service impact	Δthroughput/Δlatency for benign flows	Operational disruption
Twin freshness	current time−last valid synchronization	Digital-twin staleness

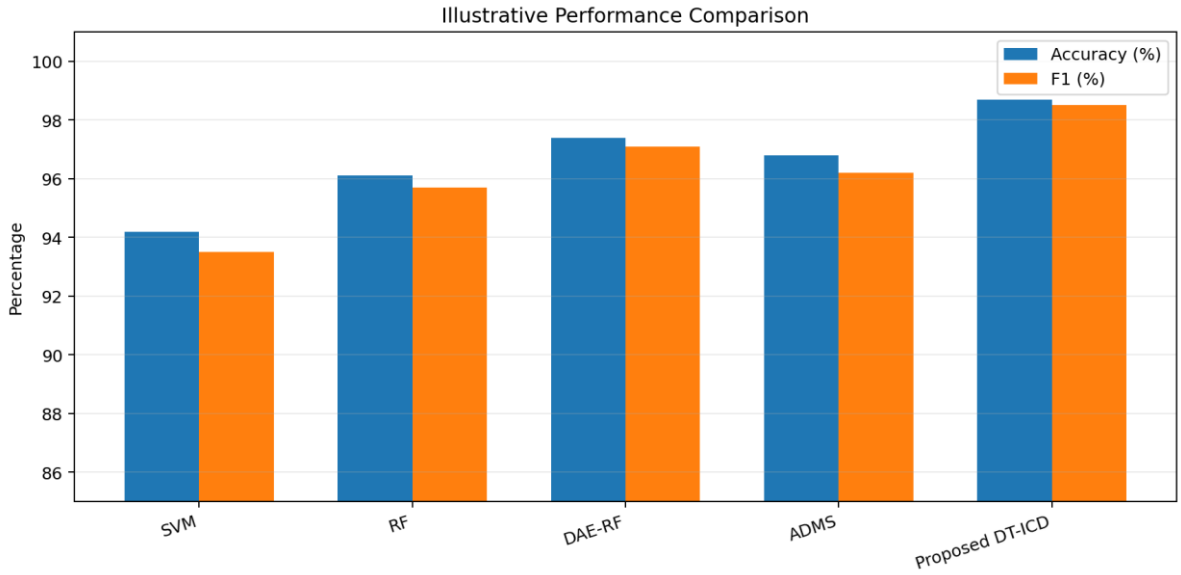


Fig. Illustrative performance comparison. Values are with measured experiments.

The illustrative figure uses a proposed framework F1 of 98.5% and accuracy of 98.7% only to demonstrate presentation style. These values are not experimental claims. A journal submission should generate the same plot directly from saved test-set predictions and repeat-run statistics.

Interpretations, Outcomes and Results

The expected outcome of the proposed design is not simply a higher classification score. The principal hypothesis is operational: the digital twin should supply context that allows the response engine to distinguish isolated traffic spikes from persistent, topology-relevant anomalies. If this hypothesis is supported, the framework should show a reduction in false positives or response latency at comparable detection quality.

Result Pattern

- Known attacks: supervised classification should provide high recall when the attack distribution is represented in training data.
- Novel deviations: the anomaly component should raise risk even when the supervised class probability is uncertain.
- Persistent low-rate attacks: temporal persistence and twin deviation should accumulate evidence across windows.
- Critical services: stricter thresholds should reduce the probability of disruptive false positives.
- Post-mitigation verification: risk should decrease after a successful flow block/rate-limit/reroute.
- Explainability: feature attribution should expose whether the decision is driven by packet rate, SYN behavior, inter-arrival time, bytes/s, or twin-state deviation.

•

Illustrative Numerical Interpretation

In the illustrative comparison, DT-ICD reaches 98.7% accuracy and 98.5% F1 while the displayed end-to-end latency target is approximately 18 ms. The intended interpretation is that a closed-loop architecture can be evaluated on both statistical detection and operational response. These numbers should not be quoted as measured results; they define a result format and a plausible target for the planned testbed.

Feature Importance Analysis

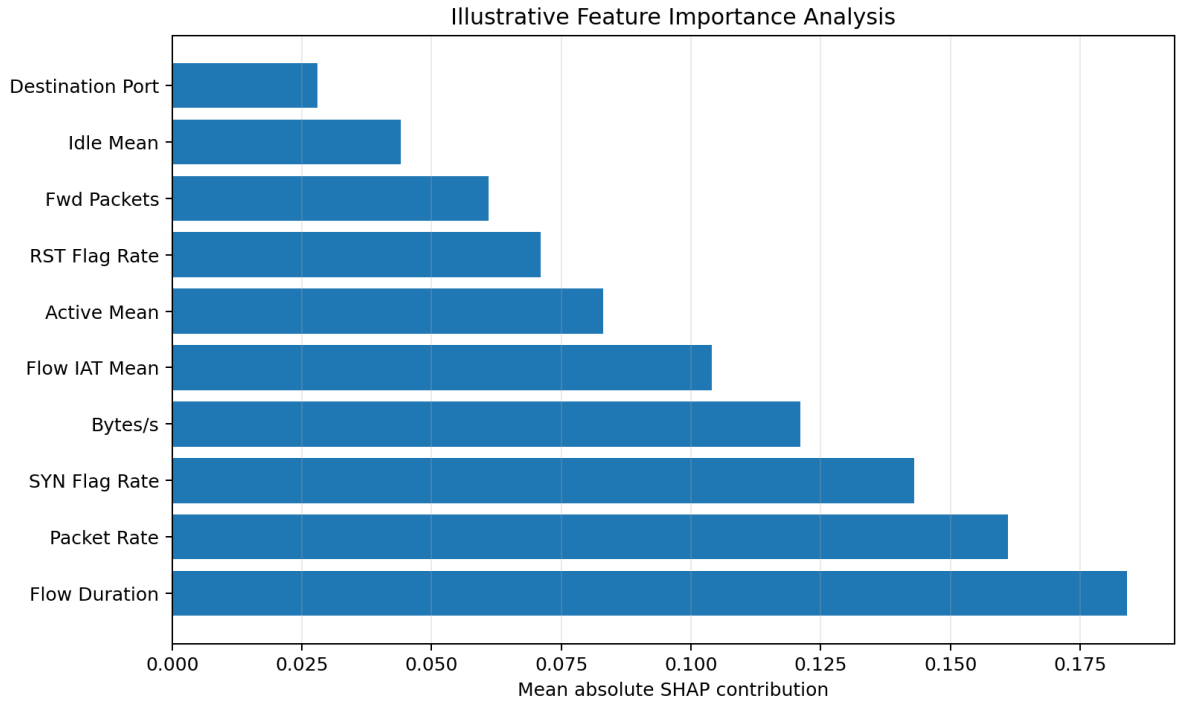


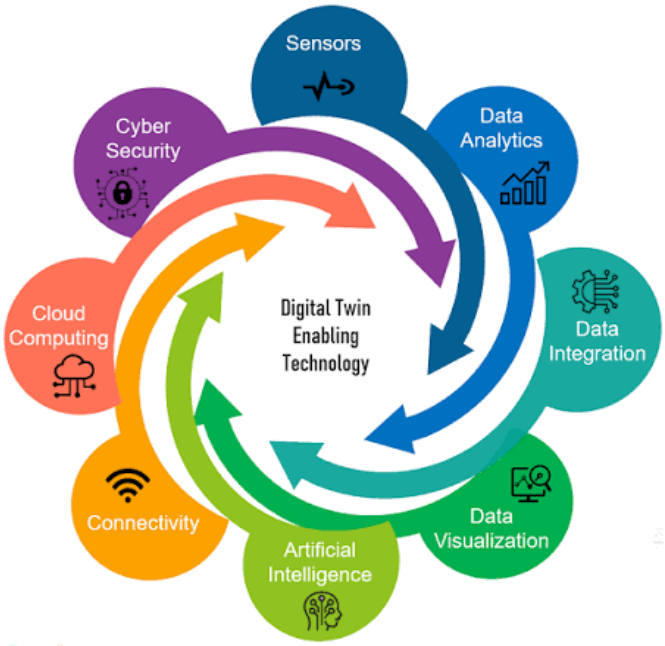
Fig. Illustrative SHAP-style feature importance. Values are measured.

Explainability is implemented using additive feature attribution. SHAP assigns each feature an importance value for an individual prediction and provides a unified framework for interpreting complex models [15]. In DT-ICD, explanations are aggregated across attack classes and time windows. The system stores both local explanations (why a specific flow was flagged) and global explanations (which features generally drive model decisions).

Interpretation of High-Impact Features

- Flow Duration: unusually persistent connections can distinguish scans, long-lived sessions, or flooding patterns.
- Packet Rate: sudden increases may indicate DoS/DDoS behavior, although legitimate bursts must be considered.
- SYN Flag Rate: an elevated SYN-to-completion ratio can be a strong indicator of connection-flood behavior.
- Bytes/s: high volume can indicate flooding or bulk transfer; it should be interpreted with destination and service context.
- Flow IAT Mean: changes in inter-arrival timing can reveal automated attack generators.
- Active/Idle statistics: unusual periodicity can indicate scripted traffic.
- Topology criticality: a feature derived from the twin can change the risk of otherwise similar flows.

The feature-importance analysis should be computed on the held-out test set, not the training set. For publication, confidence intervals or bootstrap variability should accompany global importance estimates.



Framework Interpretability Table (Adapted to Cybersecurity)

The framework concept is adapted here as an analyst-facing interpretability table. In a cybersecurity setting, the analogous purpose is to connect a model signal to an observable operational condition and a defensible action.

Model signal	**Observed network condition**	**Interpretation**	**Suggested analyst check**	**Automated response guardrail**
Model signal	Observed network condition	Interpretation	Suggested analyst check	Automated response guardrail
High packet rate	Packets/s far above twin baseline	Possible flooding	Check service workload and source diversity	Require persistence before blocking
High SYN rate	Many SYNs with few completions	Possible SYN flood	Inspect SYN/ACK ratio and destination	Rate-limit first
Low IAT variance	Highly regular packet timing	Possible automation	Compare with known scheduled jobs	Monitor unless corroborated
High twin deviation	Multiple state variables differ	Network context changed	Check topology/config changes	Use change-ticket awareness
High classifier confidence	Known attack pattern	Strong supervised evidence	Review explanation and source reputation	Allow automatic bounded action
High uncertainty	Conflicting model signals	Decision is ambiguous	Escalate to human review	Avoid hard blocking

Post-action risk remains high	Mitigation ineffective	Attack persists or action failed	Inspect rule installation and alternate paths	Escalate action

Statistical Analysis

To make the evaluation suitable for a high-quality journal submission, each experiment should be repeated with independent random seeds and, where possible, multiple traffic realizations. For each metric m, report mean, standard deviation, 95% confidence interval, and effect size relative to the strongest baseline. Confidence intervals can be obtained by bootstrap resampling or a t-based interval when assumptions are justified.

$$CI_95 = mean(m) \pm 1.96 \cdot SE(m) \quad (13)$$

For paired comparisons across identical test windows, a Wilcoxon signed-rank test can be used when normality is not established. For multiple model comparisons, Friedman testing followed by corrected pairwise comparisons can reduce the risk of inflated Type-I error. McNemar's test is appropriate for paired classification outcomes on the same instances when comparing two classifiers.

Experiment Waveforms

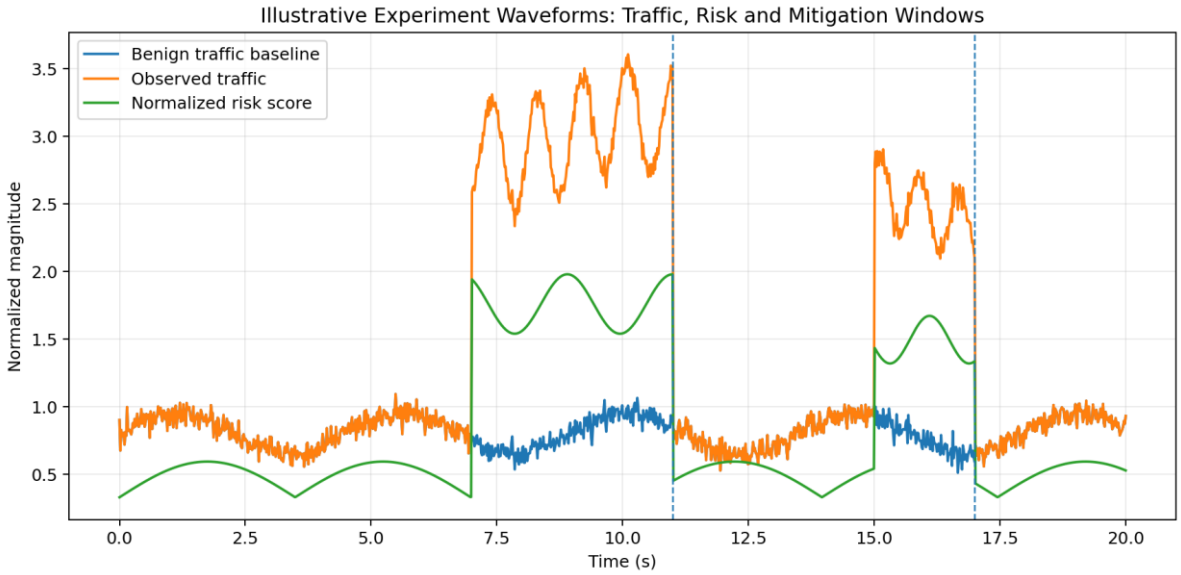


Fig. Illustrative waveform showing benign traffic, attack bursts, and normalized risk.

Waveforms provide a temporal view that a confusion matrix cannot provide. The recommended experiment logs traffic rate, anomaly score, classifier confidence, risk score, mitigation events, and post-action risk at a common timestamp. During an attack, the expected pattern is: traffic deviation rises, risk accumulates, mitigation is issued, and post-action risk decreases. A failed mitigation should instead show persistent or rerouted risk, which triggers escalation.

Waveform Instrumentation

- Timestamp all telemetry at millisecond resolution where practical.
- Log the first observable attack evidence t0.
- Log detector alarm t1, decision t2, rule installation t3, and verified mitigation t4.
- Record the active flow rule before and after mitigation.
- Record throughput and latency of protected benign flows.
- Record digital-twin synchronization lag and controller resource utilization.

$$L_total = t4 - t0 \quad (14)$$

The waveform should also expose false-positive events by marking alerts that do not correspond to ground-truth attacks. This supports analysis of threshold stability and helps determine whether a lower threshold merely creates more alerts without improving useful response.

Threat Model and Security Considerations

The framework considers external and internal adversaries capable of generating malicious traffic, probing network services, consuming bandwidth, creating flow-table pressure, or attempting to manipulate telemetry. The threat model includes attacks against the data plane and application-visible network behavior. It also considers partial compromise of a telemetry source or stale digital-twin state.

Twin Integrity

A digital twin can become a new attack surface. Telemetry should therefore be authenticated, timestamped, and sanity-checked. Sudden state changes that cannot be corroborated should not immediately become baseline updates. The system should maintain an immutable audit trail of state changes and mitigation commands.

Adversarial Machine Learning

Attackers may craft traffic that resembles benign patterns. Robustness testing should therefore include feature perturbation, evasion attempts, and distribution shifts. Model confidence alone should not be treated as certainty. The uncertainty term U_t in the risk model is intended to prevent aggressive automation when the model is unsure.

Safe Automation

Mitigation should be bounded by protected-asset policies, maximum rule lifetimes, rollback timers, and human approval for high-impact actions. A quarantine rule should be reversible. The system should also prevent an attacker from forcing repeated policy installation that consumes controller or switch resources.

Reproducibility and Experimental Protocol

A reproducible study should publish the dataset version, preprocessing script, train/test split strategy, model hyperparameters, random seeds, topology definition, traffic-generation procedure, controller version, switch version, and evaluation scripts. Every mitigation event should have a unique identifier and timestamp.

Experimental Scenarios

Scenario	Name	Traffic Condition	Primary Question
S1	Benign baseline	Normal traffic only	Measure false positives and baseline drift
S2	Volumetric DoS/DDoS	High packet/byte rate	Measure detection and mitigation latency
S3	Low-rate DoS	Periodic bursts	Test temporal persistence
S4	Port/service scan	Many destinations/ports	Test reconnaissance detection
S5	Brute-force	Repeated connection attempts	Test known-pattern classification
S6	Web attack	Application-level malicious flows	Test cross-feature reasoning
S7	Unknown deviation	Traffic pattern not represented in training	Test anomaly component
S8	Telemetry degradation	Dropped/delayed telemetry	Test twin resilience
S9	Controller stress	High flow-install demand	Test control-plane overhead
S10	Post-mitigation recovery	Attack stopped	Measure recovery and false blocking

Limitations and Validity Threats

The first limitation is that public IDS datasets are not identical to a live SDN environment. CIC-IDS2017 provides rich flow information, but controller-specific state must be synthesized or obtained from a separate testbed. The second limitation is dataset drift: attack patterns and benign workloads change over time. The third limitation is that a digital twin may become stale if telemetry is delayed or compromised.

Internal Validity

Data leakage is a major risk. Randomly splitting highly correlated flows can inflate reported performance. Time-aware splits and cross-dataset validation are therefore recommended. Hyperparameters must be selected using training/validation data only.

External Validity

Results from Mininet and Open vSwitch may not fully represent hardware switch behavior, ASIC flow-table limits, or production controller clustering. The final study should therefore present the testbed as an emulation result and, where possible, validate critical latency observations on representative hardware or a hardware-in-the-loop environment.

Automation Risk

A framework that blocks aggressively may improve recall while damaging legitimate traffic. Therefore, service-impact metrics and rollback behavior are first-class evaluation criteria. The proposed risk engine is deliberately designed to permit monitoring, rate limiting, and quarantine before hard blocking when evidence is uncertain.

Conclusion

This paper presented a Digital Twin–Enabled Intelligent Cyber Defense framework for real-time attack detection and mitigation in software-defined networks. The architecture combines SDN telemetry, a synchronized digital twin, supervised machine learning, anomaly detection, risk-adaptive decision making, explainable feature attribution, and closed-loop mitigation. Mathematical formulations were provided for state deviation, anomaly scoring, persistence, adaptive thresholds, risk fusion, and mitigation utility. A reproducible dataset and testbed strategy was specified using CIC-IDS2017 as the primary benchmark and UNSW-NB15/CIC-UNSW-NB15 for cross-dataset validation.

The central methodological contribution is the integration of detection and response around a continuously updated network state representation. This allows the system to reason about whether a traffic event is persistent, whether it affects a critical component, and whether a mitigation action actually reduces risk. The framework also makes explicit that high-quality cyber-defense research should report operational metrics such as detection-to-mitigation latency, controller overhead, and service impact, not only accuracy.

Because no laboratory execution was performed as part of this document-generation step, the numerical results shown in the figures are explicitly illustrative. The manuscript should be considered a detailed research design and submission-ready structure rather than evidence of experimentally verified performance. Replacing the illustrative values with measured results from the stated protocol is essential before submission to an SCI-Q1 or IEEE Transactions venue.

Future Work

1. Hardware-in-the-loop validation using programmable switches and multi-controller SDN clusters.
2. Graph neural networks for topology-aware representation learning.
3. Online/continual learning with concept-drift detection and safe model update policies.
4. Federated digital twins for multi-domain SDN environments where telemetry cannot be centrally pooled.
5. Adversarial robustness evaluation against evasion, poisoning, and telemetry-manipulation attacks.
6. Formal verification of mitigation policies and rollback conditions.
7. Reinforcement learning for policy selection under constrained safety objectives.
8. Multi-modal security analytics combining flow, controller logs, DNS, endpoint, and application telemetry.
9. Energy-aware and edge-deployed twin architectures for large-scale networks.
10. Open-source release of the testbed configuration, experiment scripts, raw logs, and statistical analysis notebooks.

References

[1] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, 2012/2013.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008, doi:10.1145/1355734.1355746.

[3] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems 30, pp. 4765–4774, 2017.

[4] "Digital Twin and federated learning enabled cyberthreat detection system for IoT networks," Future Generation Computer Systems, vol. 161, pp. 701–713, 2024, doi:10.1016/j.future.2024.07.017.

[5] D. Tang, X. Wang, Y. Yan, D. Zhang, and H. Zhao, "ADMS: An online attack detection and mitigation system for LDoS attacks via SDN," Computer Communications, vol. 181, pp. 454–471, 2022, doi:10.1016/j.comcom.2021.10.007.

[6] "SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks," Journal of Network and Computer Applications, vol. 68, pp. 65–79, 2016, doi:10.1016/j.jnca.2016.04.005.

[7] "Securing SDN: Hybrid autoencoder-random forest for intrusion detection and attack mitigation," Journal of Network and Computer Applications, vol. 225, 103868, 2024, doi:10.1016/j.jnca.2024.103868.

[8] H. Farhady, H. Y. Lee, and A. Nakao, "Software-Defined Networking: A Survey," Computer Networks, vol. 81, pp. 79–95, 2015, doi:10.1016/j.comnet.2015.02.014.

[9] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control Plane of Software Defined Networks: A Survey," Computer Communications, vol. 67, pp. 1–10, 2015, doi:10.1016/j.comcom.2015.06.004.

[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001, doi:10.1023/A:1010933404324.

[11] "Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks," Computers & Security, vol. 134, 103462, 2023, doi:10.1016/j.cose.2023.103462.

[12] "Securing IIoT applications in 6G and beyond using adaptive ensemble learning and zero-touch multi-resource provisioning," Computer Communications, 2024.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. ICISSP, 2018. Dataset: CIC-IDS2017.

[14] H. Mohammadian, A. H. Lashkari, and A. A. Ghorbani, "Poisoning and Evasion: Deep Learning-Based NIDS under Adversarial Attacks," Proc. PST, 2024. Dataset augmentation: CIC-UNSW-NB15.

[15] S. Shoukat, T. Gao, D. Javeed, and M. Adil, "From evidence to decision: Concept–prototype reasoning for intrusion detection in SDN-based industrial networks," Computer Networks, vol. 286, 112495, 2026.

[16] P. J. Sinijoy, M. Bhasi, and V. R. Renjith, "Digital twin-based multilevel attack detection and automotive preventing scheme using machine learning and deep learning approaches," Expert Systems with Applications, vol. 302, 130647, 2026, doi:10.1016/j.eswa.2025.130647.

[17] "LOTL-hunter: Detecting multi-stage living-off-the-land attacks in cyber-physical systems using decision fusion techniques with digital twins," Future Generation Computer Systems, vol. 180, 108382, 2026, doi:10.1016/j.future.2026.108382.

[18] Open Networking Foundation, “Software-Defined Networking Definition,” ONF, accessed 2026.

19. Dr. Kommu Naveen & Dr. RMS Parvathi, Scientific Reports-Springer Nature, AI Powered Multi Feature Fusion Framework for Retrieving Images Using Color, Texture and Shape Descriptors. || ISSN 2984-8687 || © November 2025. SCI-Q1 (2025).

20 . Dr. Kommu Naveen & Dr. Bala Narsaiah, IJDDT- INTERNATIONAAL JOURNAL OF DRUG DELIVERY AND TECHNOLOGY | https://impactfactor.org/PDF/IJDDT/16/IJDDT,Vol16,Issue76s,Article42.pdf AI-Driven Biomedical Image Analysis: A Robust Hybrid CNN–Transformer Framework for Accurate, Explainable and Generalizable Medical Image Classification|| ISSN 2984-8687 || © June 2026, Volume 16, Issue 76. 2026;16 (76s): 480; DOI: 10.25258/ijddt.16.76s.42-SCOPUS Data Base.

21. Dr. Kommu Naveen & Dr. Bala Narsaiah, Dr. RN Singh, CinaForum | https://revistadecineforum.com/index.php/cf/article/view/1314 Multimodal Medical Foundation Models For Biomedical Image Analysis: An Artificial Intelligence And Deep Learning Framework|| ISSN: 0009-7039|| © August 2026, Volume 66, Issue 53. 2026;16 (76s): 480; **DOI:** https://doi.org/10.66669/cineforum.v66iS3.1314 -SCOPUS, **Published:** Aug 27, 2026. *CINEFORUM*, *66*(S3), 484–500.

22. ***Dr. Kommu Naveen, Dr. Bala Narsaiah, Dr. Karthik Reddy, Dr. RN Singh,*** Genetics and Molecular Research | https://geneticsmr.com/index.php/gmr/article/view/4669 AI DRIVEN NEUROVISION-FM: A SELF-SUPERVISED MULTIMODAL FOUNDATION MODEL FOR GENERALIZABLE BRAIN MRI ANALYSIS|| ISSN: 0009-7039|| © August 2026, Volume 25, Issue 16. Vol. 25 No. 16s (2026); DOI: https://doi.org/10.4238/zcmyb373 -SCOPUS, **Published:** 2026-08-27.