WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

NeuroNova: An Adaptive AI-Based Cyber Defense Framework for Behavioral Intrusion Analysis, Dynamic Deception, and Automated Containment

Siddhesh Chavan[1], Atharva Shinde[2], Parth Pandit[3], Prajkta Puranik[4]

[1,2,3]Department Of Artificial Intelligence And Machine Learning Engineering, ISBM College Of Engineering, Pune, Maharashtra, India.
[4]Professor, Department Of Artificial Intelligence And Machine Learning Engineering, ISBM College Of Engineering, Pune, Maharashtra, India.

ABSTRACT :

Web application security controls often rely on a simple allow-or-deny decision for each request. Such controls may flag legitimate bursts of activity, reveal detection through immediate blocking, and provide limited insight into an adversary. This paper introduces NeuroNova, an adaptive defense prototype that uses session context, cumulative risk scoring, and staged responses. Its workflow contains eight steps: Detect, Collect Evidence, Score Risk, Decide, Respond, Observe, Learn, and Adapt. A score from 0 to 100 selects one of four states: Normal, Monitor, Active Deception, or Containment. During Active Deception, suspicious operations such as fund transfers are directed to a synthetic shadow environment, leaving production records unchanged. At the Containment level, the session is revoked and the source IP is temporarily blocked. Response sanitization is intended to prevent internal defensive details from reaching clients. The Threat DNA module maps observed activity to MITRE ATT&CK and derives a 16-character SHA-256-based behavioral fingerprint. The prototype was assessed in NovaBank, a synthetic retail-banking testbed, using automated tools, manual penetration scenarios, and legitimate high-frequency activity. In the reported tests, 30 consecutive history checks by a legitimate user retained a risk score of zero; every included injection and enumeration category was detected; deception controls behaved as designed; production records remained isolated during shadow processing; and all 72 automated tests passed. These findings are limited to the controlled environment and payloads examined, and should not be interpreted as proof of universal detection or production readiness.

Keywords: adaptive cyber defense, cyber deception, honeypot systems, behavioral analysis, risk-based response, automated containment, MITRE ATT&CK, web security

1. Introduction

Financial web applications face a broad range of threats, including injection attacks, credential stuffing, attempts to gain unauthorized privileges, and automated reconnaissance. Many conventional controls, including web application firewalls and request-rate limits, evaluate requests through a largely binary rule: permit or deny. Although useful, this approach has important shortcomings.

Fixed thresholds can classify legitimate bursts of activity as hostile, even when a customer is repeatedly accessing their own records. A hard block can also signal that a request has been detected, giving an attacker a reason to alter their approach. Finally, a simple denial often provides limited opportunity to observe an adversary's behavior and use that evidence to improve later decisions.

Cyber deception supplements prevention by presenting carefully isolated decoys, such as simulated endpoints, canary credentials, and fabricated transaction records. Interactions with these resources can provide useful signals about an intruder without exposing genuine assets. While honeypots are an established security technique, they are not always connected to session-level risk assessment and automated response within the same application workflow.

This work introduces NeuroNova, a defensive platform that brings these capabilities together. Its principal contributions are:

1. A feedback-driven defense workflow that evaluates activity across a session rather than treating each request as an unrelated event.
2. A deterministic policy engine that translates a session risk score from 0 to 100 into four response states: Normal, Monitor, Active Deception, and Containment.
3. A deception layer combining decoy endpoints, canary accounts, hidden form fields, and a shadow banking environment separated from production PostgreSQL records.
4. A response-sanitization policy designed to keep internal risk values, detection flags, and defensive terminology out of client-visible output.
5. A Threat DNA module that associates observed behavior with MITRE ATT&CK techniques and creates 16-character SHA-256-derived behavioral fingerprints for presentation in a security operations dashboard.
6. An evaluation using the synthetic NovaBank environment and an automated test suite containing 72 tests.

The remainder of this paper is structured as follows: Section 2 discusses related research; Section 3 explains the architecture; Section 4 describes implementation; Sections 5 and 6 present the test design and results; Section 7 examines limitations; and Section 8 concludes the paper.

2. Related Work

Honeypots and cyber deception. Honeypots are deliberately exposed systems or resources intended to attract and record suspicious interaction. Spitzner (2003) described their use for tracking intruders, while Provos (2004) presented a virtual framework capable of emulating multiple hosts. Rowe and Rrushi (2016) discuss deception as a broader defensive practice. NeuroNova adapts these principles to web applications and connects decoy interaction to a continuously updated session risk score instead of operating the traps as independent tools.

Decoy credentials and honeywords. Juels and Rivest (2013) introduced honeywords: false passwords placed alongside genuine password data so that use of a decoy can reveal possible compromise. NeuroNova applies a related idea through canary accounts. An attempt to authenticate with one of these credentials is treated as a strong indicator of suspicious activity.

Signature detection and behavioral analysis. Pattern-based controls can identify known attack strings, including SQL injection and cross-site scripting, both addressed in OWASP's web application security guidance (OWASP Foundation, n.d.). Pattern matching alone may not capture the wider context of a session. NeuroNova therefore combines payload inspection with information about identity, authorization, and the resources a user is entitled to access.

Adversary modeling. MITRE ATT&CK organizes documented adversary tactics and techniques into a common reference framework (Strom et al., 2018). Linking observed activity to this framework gives analysts a consistent way to describe attack behavior. NeuroNova uses these mappings in its profiling and reporting functions.

Research position. NeuroNova's central design choice is to combine contextual detection, graduated risk decisions, shadow-environment redirection, sanitized client responses, and ATT&CK-based behavioral profiling in a single feedback loop.

3. System Architecture

3.1 Feedback-Driven Defense Cycle

NeuroNova processes activity through eight connected stages:

DETECT → COLLECT EVIDENCE → SCORE RISK → DECIDE → RESPOND → OBSERVE → LEARN → ADAPT

For each request, the service inspects the activity and associates relevant evidence with the current session. It recalculates the accumulated risk, selects the corresponding policy state, and applies the required action. The observed result then informs behavioral profiling and subsequent risk decisions.

3.2 Main Components

Table 1 lists the platform modules and summarizes each module's responsibility.

Table 1. NeuroNova modules and responsibilities.

Component	Responsibility
Detector (detector.py)	Pattern-based checks for SQL injection, XSS, path traversal, command injection, and brute-force activity
Behavioral context layer	Uses identity and authorization context to identify IDOR, parameter manipulation, and API enumeration
Policy engine (policy_engine.py)	Converts the accumulated risk score into a response state and applies containment
Honeypot subsystem (honeypot.py)	Provides decoy routes, canary accounts, and hidden-field traps
Shadow decoy engine (shadow_decoy.py)	Processes suspicious transfers against a synthetic ledger
Threat DNA engine (threat_dna.py)	Associates observed behavior with ATT&CK and generates behavioral fingerprints
SOC dashboard	Displays live security events and supports manual containment override
Datastores	PostgreSQL for application records; Redis for expiring IP-block entries

3.3 Risk Policy and Response States

The policy engine uses the accumulated session score to select one of the defined response states. The score bands and their associated actions are shown in Table 2.

Table 2. Session risk bands and corresponding actions.

Risk score	State	Action
0–59	D0_NORMAL	Allow standard access

60–79	D1_MONITOR	Increase logging and restrict sensitive endpoints
80–89	D4_ACTIVE_DECEPTION	Redirect selected traffic to decoys and the shadow environment
≥ 90	D5_CONTAINMENT	Revoke the session and temporarily block the source IP in Redis

Since each state is selected using fixed score boundaries, the same score produces the same policy outcome. This predictable behavior supports repeatable testing and makes the policy easier to audit.

3.4 Deception Layer

Decoy endpoints. The honeypot service offers artificial resources that imitate items such as audit records, configuration files, database dumps, and administrator interfaces. These resources are kept separate from real production assets.

Canary accounts. Attempts to sign in with decoy credentials, including names such as admin, root, bank_manager, and db_admin, contribute 35 points to the session score and are assigned a stated confidence of 95%. The client receives a generic authentication failure rather than a message revealing that the account is a trap.

Hidden-field traps. The application includes CSS-hidden form fields, for example website and company_fax, that ordinary users are not expected to complete. If a field is populated, the system adds 40 risk points, assigns 99% confidence, and labels the activity as credential-bot behavior.

Shadow banking environment. When a session enters state D4, a suspicious wire transfer is processed against a synthetic ledger. A plausible receipt and transaction identifier are returned, but actual PostgreSQL balances and ledger records remain unchanged.

3.5 Containment and Client-Facing Information Controls

When the score reaches 90, NeuroNova terminates the session and places the source IP on a Redis blocklist with a time-to-live, allowing the entry to expire automatically. Blocked requests receive a consistent HTTP 503 message: “Bank service is temporarily unavailable. Please try again later.” Output sanitization is intended to keep internal details, including scores, deception flags, and honeypot markers, out of response bodies, headers, and browser-visible content.

3.6 Threat DNA and Security Operations

Threat DNA examines HTTP-level indicators associated with tools such as SQLMap, Nikto, Gobuster, cURL, Hydra, and Python Requests. It maps detected behavior to MITRE ATT&CK and creates a 16-character fingerprint derived from SHA-256. Because this identifier is based on behavioral signals rather than solely on an IP address, it can help associate similar activity across sessions and changing addresses. The security operations center (SOC) dashboard displays telemetry and allows an operator to override containment decisions.

4. Implementation

The platform is built as a Python web service. PostgreSQL stores the test application's primary records, while Redis maintains temporary state such as IP blocks. The NovaBank testbed simulates a retail banking site with time-based one-time password (TOTP) multi-factor authentication, account management, transaction history, and transfers. Policy checks are applied at the request entry point so that a containment decision can take effect before the request reaches the application's core logic.

The ATT&CK relationships implemented in the current prototype are summarized in Table 3.

Table 3. Mapping from observed behavior to MITRE ATT&CK techniques.

Observed behavior	ATT&CK technique
SQL injection	T1190, Exploit Public-Facing Application
Command injection	T1059, Command and Scripting Interpreter
Path traversal / LFI	T1083, File and Directory Discovery
Password guessing	T1110.001, Brute Force: Password Guessing
Canary account probing	T1078, Valid Accounts
Shadow transfer fraud	T1565.001, Stored Data Manipulation

5. Experimental Setup

Testing was conducted in the NovaBank environment using three broad traffic groups:

1. **Automated security tools, including SQLMap, Nikto, Gobuster, cURL, Hydra, and Python Requests.**
2. **Manually executed penetration scenarios involving injection, path traversal, endpoint enumeration, canary-account attempts, and fraudulent transfers.**
3. **Benign but frequent user activity, including repeated access to personal transaction history and authorized changes to account credentials.**

An automated pytest suite of 72 cases was used to check individual modules, deception behavior, policy thresholds, and end-to-end integration. The tested payload families included the following:

- SQL injection: tautological conditions such as ' OR '1'='1, UNION SELECT statements, time-delay probes using SLEEP(), and stacked commands.
- Cross-site scripting (XSS): inline script content, event-handler payloads such as onerror and onload, and DOM-oriented injection attempts.
- Path traversal: attempts using sequences such as ../../../../etc/passwd and Windows system paths.
- Command injection: shell control characters, subshell expressions such as $(whoami), and pipe-based command chaining.
- Brute-force and credential-stuffing activity: repeated authentication attempts that exceeded a sliding-window threshold.

6. Results

6.1 Evaluation Overview

Table 4 compares the tested scenarios with the expected behavior of a conventional defensive baseline and records NeuroNova's observed response.

Table 4. Results across the evaluated scenarios.

Scenario	Test activity	Conventional baseline	NeuroNova response	Observed outcome
Legitimate user activity	30 history checks and profile updates	Static thresholds may generate false alarms	Score remained 0 (D0_NORMAL)	No false-positive escalation in this test
Authentication defense	Attempts with canary names such as admin, root, and bank_manager	Potential unnoticed account probing	Trap added 35 points; stated confidence 95%	Generic 401 response; no trap disclosure
Bot mitigation	Submission of a hidden form field	May be treated as an ordinary submission	Trap added 40 points; stated confidence 99%	Classified as CREDENTIAL_BOT
Web attack payloads	SQLi, XSS, traversal, and command injection	Potential data exposure or command execution	Included payloads detected at ≥90% confidence where reported	Events mapped to ATT&CK techniques
Active deception (D4)	Suspicious wire transfer	Risk of modifying real balances	Operation sent to shadow ledger	Production database unchanged
Containment (D5)	Attack burst with score ≥90	Response may require manual action	Session revoked and IP placed on Redis blocklist	Generic 503 response without internal terms
Behavioral profiling	HTTP tool fingerprinting	Session may remain unattributed	16-character Threat DNA fingerprint generated	Behavior linked to ATT&CK

6.2 Handling Legitimate High-Frequency Activity

During the integration test, a legitimate account accessed its own transaction history 30 times in succession and completed authorized credential updates. The risk score remained at zero, leaving the session in D0_NORMAL. The system did not treat request frequency as sufficient evidence of an attack because the identity and authorization context confirmed access to the user's own records. This differs from a rate-limit-only approach, where frequent requests can trigger suspicion without considering ownership.

6.3 Detection of Tested Attack Payloads

The test suite detected each attack category included in the evaluation: SQL injection, XSS, path traversal, command injection, and brute-force activity. The reported confidence for SQL injection and XSS alerts was at least 90%. Repeated-login attacks were identified by the sliding-window mechanism after the configured limit was crossed.

6.4 Canary and Hidden-Field Signals

The canary-account signal contributed 35 risk points at a reported confidence of 95%; the hidden-field signal contributed 40 points at 99% confidence. These controls are designed to provide high-specificity evidence because normal customers should neither submit decoy login credentials nor fill in fields hidden from ordinary interaction.

6.5 Isolation of Production Records During Deception

For sessions in state D4, with scores between 80 and 89, suspicious transfers were handled by the shadow ledger. The simulated attacker received a transaction receipt containing a plausible identifier, while the production database's actual account balances and ledger entries were left intact.

6.6 Repeatable Containment and Response Sanitization

Policy tests confirmed the expected state changes at the 59/60, 79/80, and 89/90 boundaries, without observed threshold instability. When the score reached at least 90, the session was revoked and its source address was added to the Redis blocklist. All tested blocked responses used the same generic HTTP 503 message. Inspection of the tested output found none of the internal terms "honeypot," "attacker," "risk_score," "blocked," or "contained."

6.7 Behavioral Fingerprinting and ATT&CK Mapping

In the tested scenarios, the profiling module recognized HTTP fingerprints associated with SQLMap, Nikto, Gobuster, cURL, Hydra, and Python Requests. It assigned the corresponding ATT&CK techniques listed in Table 3 and generated a repeatable 16-character fingerprint for each behavioral profile.

6.8 Automated Test Results

The reported pytest run completed all 72 tests successfully, representing a 100% pass rate in 87.66 seconds. The module-level breakdown is provided in Table 5.

Table 5. Results from the automated test suite.

Module	Coverage	Passed
test_policy_engine.py	Policy logic and score boundaries	7/7
test_two_factor.py	Two-factor authentication flow	6/6
test_honeypot.py	Attack indicators and decoy routing	24/24
test_shadow_decoy.py	Isolation of shadow-ledger operations	6/6
test_threat_dna.py	Behavioral fingerprinting and ATT&CK mapping	5/5
test_user_journey.py	End-to-end journeys and false-positive checks	6/6
test_soc_endpoints.py	SOC functions and containment override	6/6
Total		72/72

7. Discussion and Limitations

The prototype results indicate that session context, graduated response states, and deception can work together to reduce unnecessary escalation during the tested legitimate journeys while collecting more information than a simple deny response. The findings should nevertheless be interpreted within the following limits:

- **Synthetic environment. NovaBank is a test application rather than a production banking service. Its results may differ from those seen under real operational traffic, larger workloads, or different application designs.**
- **Restricted attack coverage. The reported full detection result applies only to the payload families and tools included in these experiments. It is not evidence of universal detection: encoding, obfuscation, and other adaptive techniques may evade signature-oriented checks.**
- **Limited benign-user sample. The false-positive observation is based on predefined user journeys, not a large and varied group of real customers. Broader testing with measured precision, recall, and false-positive rates is needed.**
- **Passing tests does not establish complete security. The 72 cases check specified behavior, but cannot demonstrate that every bypass or implementation weakness has been eliminated.**
- **Potentially detectable deception. Experienced adversaries might identify synthetic responses. The zero-leakage checks covered specified terms and response channels, not every possible side channel, including timing.**
- **Fingerprint reliability at scale. Cross-IP association was demonstrated with the test profiles used here. Collision rates, resistance to evasion, and performance at larger scale require additional study.**

8. Conclusion and Future Work3

NeuroNova is a web-application defense prototype that combines behavioral intrusion analysis, adaptive deception, and automated containment in a feedback-driven process. In the NovaBank test environment, the evaluated legitimate high-frequency journey did not trigger a risk increase; the included attack categories were detected; deceptive transfers did not alter production records; policy transitions followed their defined thresholds; and the inspected client responses did not expose the specified internal security terms.

Further work should test encoded and evasive payloads, compare the platform with established web application firewall configurations such as ModSecurity, measure latency and throughput costs, and evaluate the system with larger and more diverse traffic datasets. Additional research should also investigate learned risk-scoring components, possible side-channel leakage, and the stability of behavioral fingerprints under deliberate evasion.

REFERENCES

1. Var Naseri, M., Welch, I., Haseeb, J., & Mansoori, M. (2026). Q-Cowrie: An adaptive honeypot to analyse attackers' behaviour. International Journal of Information Security, 25(2), Article 60. https://doi.org/10.1007/s10207-026-01221-5
2. Hoang, H. D., To, T.-N., Son, N. D. H., Ngo-Khanh, K., Cam, N. T., & Pham, V.-H. (2026). Hawkeyes: An intelligent honeypot allocation strategy for cyber deception using reinforcement learning. Computer Networks, 276, Article 111982. https://doi.org/10.1016/j.comnet.2025.111982
3. Beltrán-López, P., Gil Pérez, M., Vasilomanolakis, E., & Nespoli, P. (2026). Reactive cyber deception: Stealth-based adaptive redirection to on-demand honeypots with AI-driven data generation. Computer Networks, 281, Article 112203. https://doi.org/10.1016/j.comnet.2026.112203
4. Varadarajan, A., Chandrasekaran, A., Binumohan, R., Ravishankar, R. H., & Sadasivam, G. K. (2026). Intelligent honeypot for web applications: Leveraging Seq2Seq and reinforcement learning. ECTI Transactions on Computer and Information Technology. https://doi.org/10.37936/ecti-cit.2026202.262099
5. Zhang, M., Yu, Y., Li, S., Liu, Y., Zhang, Y., Zhang, R., & Shao, S. (2026). Adaptive software-defined honeypot strategy using Stackelberg game and deep reinforcement learning with DPU acceleration. Modelling, 7(1), Article 23. https://doi.org/10.3390/modelling7010023
6. Shahzad, M., & Saleh, M. H. (2026). Microservice-oriented cyber deception platform with containerized honeypots and real-time telemetry. Journal of Cybersecurity and Privacy, 6(4), Article 117. https://doi.org/10.3390/jcp6040117
7. Weinberg, A. I. (2026). PHANTOM: Polymorphic honeytoken adaptation with narrative-tailored organisational mimicry contextually convincing cyber deception at scale. International Journal of Information Security, 25, Article 154. https://doi.org/10.1007/s10207-026-01323-0
8. Dumisani, B. M., & Ould-Slimane, H. (2025). Game theory: Cyber deception based on the redundancy of diversified honeypots. Procedia Computer Science, 265, 107–115. https://doi.org/10.1016/j.procs.2025.07.162
9. Kumar, L. K. S., Nethi, S. R., Uyyala, R., et al. (2026). Anomaly-based intrusion detection on benchmark datasets for network security: A comprehensive evaluation. Scientific Reports, 16, Article 8507. https://doi.org/10.1038/s41598-026-38317-w
10. Oise, G. P., Mintah, E., Sokoya, O., Uloko, F. O., Jessa, T., Konyeha, S., et al. (2026). Attention-based intrusion detection using transformer networks on UNSW-NB15. Scientific Reports, 16, Article 33862.
11. Bharathi, R. J., Arieth, R. M., Nirmala, M., et al. (2026). Transformer-based deep learning framework for network intrusion detection with context-aware traffic pattern modeling. Discover Artificial Intelligence, 6, Article 1382. https://doi.org/10.1007/s44163-026-02273-1
12. Alanazi, A. (2026). A domain-agnostic explainable framework for network attack detection across diverse traffic datasets. Scientific Reports, 16, Article 24512.
13. Raj, S. A. S. J., & Madiajagan, M. (2026). TAE-IDS: A trust-aware explainable intrusion detection framework using attention-based meta-ensemble learning with blockchain validation. Frontiers in Artificial Intelligence, 9. https://doi.org/10.3389/frai.2026.1903112
14. Alamro, H., Alahmari, S., Nemri, N., Aljebreen, M., Alhashmi, A. A., Alamro, S., et al. (2025). Enhanced intrusion detection in cybersecurity through dimensionality reduction and explainable artificial intelligence. Scientific Reports, 15, Article 33848. https://doi.org/10.1038/s41598-025-06761-9
15. Milani, A., Franzoni, V., & Florindi, E. (2026). Indirect prompt injection in large language models. Neural Computing and Applications, 38, Article 530. https://doi.org/10.1007/s00521-026-12266-x
16. Geng, T., Xu, Z., Qu, Y., & Wong, W. E. (2026). Prompt injection attacks on large language models: A survey of attack methods, root causes, and defense strategies. Computers, Materials & Continua. https://doi.org/10.32604/cmc.2025.074081
17. Alzahrani, A. (2026). PromptGuard: A structured framework for injection-resilient language models. Scientific Reports, 16, Article 1277. https://doi.org/10.1038/s41598-025-31086-y
18. Gulyamov, S., Gulyamov, S., Rodionov, A., Khursanov, R., Mekhmonov, K., Babaev, D., & Rakhimjonov, A. (2026). Prompt injection attacks in large language models and AI agent systems: A comprehensive review of vulnerabilities, attack vectors, and defense mechanisms. Information, 17(1), Article 54. https://doi.org/10.3390/info17010054
19. Habibzadeh, et al. (2026). Large language models for security operations centers: A comprehensive survey. Journal of Electrical and Computer Engineering, 2026, Article 3383674. https://doi.org/10.1155/jece/3383674
20. 20. Yu, H., Iqbal, S., Pinto Neto, E. C., Buffett, S., et al. (2026). From static tasks to dynamic reasoning: A characterization framework and study of large language models in next-generation cybersecurity automation. International Journal of Information Security, 25, Article 149.