Priyanshu, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Self-Healing Cyber Defence Using Reinforcement Learning

Priyanshu, Sakchi, Rohit Chauhan, Shivam Chauhan, Rohan Kalsa, Dr. Pratap Singh
Department of Computer Science and Engineering (AIML)
Quantum University, Roorkee

Abstract- The rapid evolution of cyber threats demands security systems capable of not only detecting intrusions but autonomously responding and recovering without human intervention. This paper presents a Self-Healing Cyber Defence System leveraging Deep Q-Network (DQN) Reinforcement Learning to enable adaptive, automated threat response in dynamic network environments. The proposed system employs a DQN agent trained within a custom OpenAI Gym simulation environment, enabling it to learn optimal defensive strategies against brute force, Distributed Denial of Service (DDoS), port scanning, and SQL injection attacks. Real-time network traffic analysis, state-action modelling, and reward-based learning produce an agent capable of selecting mitigation actions—including node isolation, traffic rate limiting, honeypot redirection, and system rollback—within milliseconds of detection. The system is built using Python, TensorFlow, FastAPI, SQLite, and a JavaScript-based real-time security dashboard. Testing across 10,000 simulated attack events demonstrates 94.2% detection accuracy, 89.7% successful mitigation rate, and 97.3% self-healing recovery rate, substantially outperforming static rule-based baselines. The system bridges the critical gap in existing cybersecurity infrastructure by providing an intelligent, adaptive, and autonomous defence mechanism that unifies detection, response, and recovery within a single interpretable framework.

Keywords— Reinforcement Learning, Deep Q-Network, Intrusion Detection, Self-Healing, Autonomous Cyber Defence, DDoS, Brute Force, Network Security, FastAPI, OpenAI Gym.

I. INTRODUCTION

Modern digital infrastructure faces an ever-expanding landscape of sophisticated cyber threats. Enterprise networks, cloud platforms, and IoT ecosystems are continuously targeted by adversaries employing novel, adaptive attack strategies that evolve faster than traditional security mechanisms can respond. Static, signature-based intrusion detection systems (IDS) such as Snort and Suricata remain foundational tools, yet they are fundamentally reactive—they require manual signature updates and cannot autonomously recover from successful breaches [1]. This limitation is critical in environments where attack dwell time can span weeks or months, enabling adversaries to achieve lateral movement, privilege escalation, and data exfiltration before any defensive measure is triggered. According to the Verizon 2024 Data Breach Investigations Report, the median time for organizations to detect a breach remains unacceptably high, highlighting the urgent need for machine-speed autonomous defence [17].

Reinforcement Learning (RL) offers a paradigm-shifting approach to this problem. By framing cyber defence as a sequential decision-making problem, RL agents observe network states, select defensive actions, and receive reward signals that guide the learning of optimal security policies. Unlike supervised machine learning models that require large labelled datasets, RL agents learn through direct interaction with the environment—making them uniquely suited for the dynamic, adversarial cybersecurity context where attack patterns evolve continuously [2]. The introduction of Deep Q-Networks (DQN) by Mnih et al. [3] extended RL to high-dimensional state spaces via neural network function approximators, enabling agents to reason

across complex, multi-feature network environments with significantly improved generalization compared to tabular Q-learning approaches.

This paper presents a Self-Healing Cyber Defence System that integrates a DQN RL agent with real-time network monitoring, automated threat mitigation, and autonomous system recovery into a unified, production-ready framework. The system closes the loop across all three phases of cyber incident management—detection, response, and recovery—a gap that existing RL-based security research has largely left unaddressed [4]. The system is designed with transparency and interpretability as first-class requirements: every agent decision is logged with a human-readable explanation, and a real-time security dashboard provides administrators with continuous operational visibility. The remainder of this paper is organized as follows: Section 2 reviews related literature; Section 3 defines the problem; Sections 4 and 5 present objectives and methodology; Section 6 reports results; Sections 7 and 8 discuss future work and conclusions.

II. LITERATURE SURVEY

1. Traditional Cyber Defence Mechanisms

For decades, cybersecurity systems relied on signature-based IDS that match network traffic against databases of known attack patterns. While reliable for well-documented threats, these systems are inherently reactive: novel and zero-day attacks bypass detection entirely, and signatures require continuous manual maintenance [1]. The 2003 SQL Slammer worm, which infected 75,000 hosts in ten minutes, demonstrated that human-speed signature updates are structurally insufficient against high-velocity attacks [5]. Threshold-based anomaly detection improved upon signature methods by flagging statistical deviations from baseline behaviour, yet suffered from high false positive rates and the fundamental sensitivity trade-off—conservative thresholds miss attacks while aggressive thresholds generate alert fatigue in Security Operations Centre (SOC) analysts.

IP-based blocking and account lockout policies provide simple mitigation but are easily circumvented via IP rotation, VPNs, TOR exit nodes, and distributed botnets, where each individual source remains below single-IP detection limits [6]. Security Orchestration, Automation, and Response (SOAR) platforms emerged to address manual response bottlenecks, yet they rely on pre-defined playbooks and cannot adapt to novel attack patterns without manual updates. These structural limitations—reactivity, rigidity, and the absence of autonomous recovery—create the performance gap that reinforcement learning-based approaches are architecturally positioned to address.

2. Reinforcement Learning in Cybersecurity

Early RL applications in cybersecurity focused on network intrusion detection, where agents learned to classify traffic as benign or malicious through environmental reward signals. Elderman et al. [7] demonstrated that RL agents outperform static rule-based defenders in adaptive network attack scenarios, particularly when adversarial agents were simultaneously adapting their strategies. Malialis and Kudenko [8] applied multi-agent RL to DDoS mitigation, showing that decentralized coordinated responses across router nodes significantly reduced service disruption compared to centralized IP blocking. The Microsoft Research CyberBattleSim platform [9] provided a structured simulation environment for training RL agents against multi-step network compromise scenarios modelled on real-world attacker tactics, where agents discovered non-obvious defensive strategies that human operators rarely considered—such as proactively isolating low-value nodes to deny attacker lateral movement paths.

Nasir et al. [10] proposed an autonomous intrusion response system using Q-learning, demonstrating that learned response policies significantly outperformed expert-designed rules in minimising service disruption during containment. Standen et al. [4] introduced hierarchical RL for multi-stage attack response, enabling agents to reason across both tactical and strategic defensive layers. Despite these advances, a common limitation persists: most RL-based security research addresses either detection or response in isolation. The full detect-respond-recover lifecycle—particularly the automated system

recovery phase—remains largely unaddressed in the literature, representing the primary research motivation for this work.

3. Identified Research Gaps

A comprehensive review of existing literature reveals five critical gaps: (i) absence of end-to-end self-healing—no existing RL-based security system unifies detection, response, and autonomous recovery; (ii) limited transparency—most deep learning security models operate as black boxes with no interpretable decision logic, reducing administrator trust and auditability [11]; (iii) absence of integrated dashboards—research prototypes rarely include real-time visualization tools suitable for SOC deployment; (iv) narrow attack coverage—most RL agents are trained against a single attack type, limiting generalizability across brute force, DDoS, injection, and scanning threats simultaneously; and (v) high deployment costs—existing autonomous solutions are enterprise-grade commercial products, inaccessible to SMEs and academic institutions [12]. This paper directly addresses all five gaps within a single unified, open-architecture system.

III. PROBLEM IDENTIFICATION

Contemporary cyber defence faces three fundamental challenges that static and conventional ML-based systems have collectively failed to resolve. First, reactive architecture: signature and threshold-based systems respond only to known patterns, leaving organizations vulnerable during the critical window between novel attack emergence and defensive response deployment. This window—during which attackers may achieve full system compromise—has no structural solution within rule-based frameworks. The attack surface of modern distributed systems is too large, and attacker tooling too adaptive, for static defences to provide adequate coverage.

Second, the absence of autonomous recovery: even where modern systems successfully detect and contain threats, the restoration of compromised services, databases, and configuration states requires substantial manual effort. Mean Time to Recovery (MTTR) following security incidents frequently spans hours to days, during which organizations face operational disruption, financial loss, and reputational damage. Current RL-based security research frameworks address detection and response as isolated problems, leaving the recovery dimension structurally unaddressed [4]. A complete autonomous defence architecture must treat the incident lifecycle—detect, respond, recover—as an inseparable whole.

Third, opacity of decision logic: deep learning-based security models, despite demonstrating superior classification accuracy, operate as black boxes that produce decisions without providing interpretable explanations. This opacity creates practical deployment barriers: security administrators cannot audit automated decisions, forensic analysts cannot reconstruct decision rationale for incident reports, and compliance teams cannot justify AI-driven security actions to regulators [13]. The interpretability deficit is particularly acute for autonomous systems making consequential decisions—such as blocking IP addresses or isolating network nodes—where accountability requires that every action be traceable to observable triggering conditions.

Objectives

Primary Objectives

The primary objective of this research is to design and implement a reinforcement learning-based cyber defence agent capable of autonomous threat detection, intelligent response selection, and system recovery without human intervention. The system must achieve real-time inference latency below 50 milliseconds, enabling defensive actions to outpace human-speed attack execution. The agent must demonstrate generalizability across multiple attack categories—brute force, DDoS, SQL injection, and port scanning—within a single trained policy. Additionally, the system must address the interpretability gap in deep learning security systems by generating transparent, human-readable decision logs for every agent action, supporting both administrative oversight and forensic auditability in compliance-sensitive environments.

Secondary Objectives

Secondary objectives include: (i) developing a custom OpenAI Gym simulation environment for DQN agent training against realistic multi-vector attack scenarios with configurable attack intensity and topology complexity; (ii) implementing OS-level threat mitigation through automated firewall rule enforcement via netsh (Windows) and iptables (Linux) and process isolation mechanisms; (iii) building a self-healing recovery module for atomic rollback of compromised system components to verified clean states; (iv) constructing a real-time security dashboard with live alert feeds, geographic threat mapping via Leaflet.js, and RL training curve visualization via Chart.js; (v) enabling email and in-dashboard alert notifications for critical threats; and (vi) maintaining a modular, extensible architecture to support future multi-agent RL, federated learning, SIEM integration, and cloud-native deployment.

IV. PROPOSED METHODOLOGY

1. System Overview

The Self-Healing Cyber Defence System is designed as a modular, service-oriented security framework that continuously monitors network activity, makes autonomous defensive decisions, executes mitigation actions, and restores compromised system components without human intervention. The system operates as a sense-decide-act-recover cycle, analogous to the Observe-Orient-Decide-Act (OODA) loop used in military and security operations. Each complete cycle—from event ingestion through mitigation execution to database logging—completes within a target window of 100 milliseconds for standard attack events. The core technology stack consists of Python 3.10+, TensorFlow 2.12 (DQN training and inference), OpenAI Gym 0.26 (simulation environment), FastAPI 0.100 (backend API), SQLite 3 (persistence), Scapy (packet analysis), MaxMind GeoIP2 (IP enrichment), and a JavaScript/HTML5/CSS3 frontend with Chart.js and Leaflet.js.

2. System Architecture

The architecture integrates six functional layers operating in a continuous pipeline. The Network Event Collection Layer receives real-time event data via validated FastAPI POST endpoints, enriching each payload with GeoIP geolocation data before downstream processing. The State Representation Engine aggregates event features over a configurable sliding time window, computing velocity metrics, frequency counts, geolocation risk scores, and historical threat context to construct normalized state vectors in the range [0, 1]. The DQN Inference Engine evaluates state vectors against the learned Q-value function and selects the action with the highest predicted cumulative reward within the 50-millisecond latency budget. The Mitigation and Recovery Layer executes OS-level or network-level commands corresponding to the selected action. The Persistence Layer maintains an audit trail of all events, agent decisions, Q-values, and recovery operations in SQLite. The Visualization Layer exposes all system data through REST endpoints consumed by the real-time administrative dashboard, refreshing at 5-second intervals.

3. DQN Agent Design

The DQN agent is implemented as a feed-forward neural network with two hidden layers (64 and 32 neurons, ReLU activation) using TensorFlow 2.x Keras. The state vector encodes eight features: failed login count in a 60-second sliding window, connection frequency per source IP, distinct usernames probed per IP, geolocation risk score (0–1 based on country threat intelligence), historical threat level of the source IP, current CPU load, active connection count, and packet rate—all normalized to [0, 1]. The action space consists of seven discrete actions: NO_ACTION, RATE_LIMIT, IP_BLOCK_TEMPORARY (30-minute block), IP_BLOCK_PERMANENT, NODE_ISOLATION, HONEYPOT_REDIRECT, and SYSTEM_ROLLBACK. The reward function assigns +10 for successful mitigation of a confirmed attack, −5 for false positives (blocking legitimate traffic), −10 for missed attacks that escalate, and +3 for system availability preservation. Training employs epsilon-greedy exploration (ε decaying from 1.0 to 0.01 over 500 episodes), experience replay with a replay buffer of 10,000 transitions, and a target network updated every 100 steps to provide training stability and prevent divergence.

4. Detection Parameters

Table 1: Threat Detection Parameters and Corresponding Agent Actions

Attack Type	Detection Condition	Agent Action	Purpose
Brute Force	≥6 failed logins within 60 s from one IP	IP_BLOCK_TEMPORARY	Halt credential attack
DDoS Flood	≥1000 req/sec; high packet rate anomaly	RATE_LIMIT + NODE_ISOLATION	Contain volumetric attack
Port Scan	Sequential port sweep across ≥20 ports	HONEYPOT_REDIRECT	Detect & trap reconnaissance
SQL Injection	Malicious payload pattern in request body	IP_BLOCK_PERMANENT	Prevent data exfiltration
Multi-Username Probe	≥4 distinct usernames probed from 1 IP	IP_BLOCK_PERMANENT	Detect credential enumeration
Zero-Day Anomaly	State vector beyond learned normal bounds	SYSTEM_ROLLBACK	Contain novel threats

5. Self-Healing Recovery Module

The self-healing module implements a two-phase commit recovery protocol. Upon selection of the SYSTEM_ROLLBACK action—or detection of an anomalous component state by the continuous health monitor—the module (i) identifies the affected service, configuration file, or database using a component registry maintained in SQLite; (ii) retrieves the last verified clean snapshot, identified by a SHA-256 content hash stored at snapshot time; (iii) stages the restoration to a temporary location and verifies the restored component's hash against the stored reference; and (iv) atomically commits the restoration, replacing the compromised component only upon hash verification success. Failed rollbacks are logged as critical alerts and escalated to the administrator without committing potentially corrupt state. All recovery operations are logged with timestamps, component identifiers, rollback source hashes, and operator notifications for forensic accountability.

6. Transparency and Explainability

Every agent action is accompanied by a structured decision summary documenting: (i) the triggering state vector values and which features exceeded detection thresholds; (ii) the selected action, its associated Q-value, and the Q-values of all alternative actions (providing counterfactual visibility); (iii) the expected defensive outcome; and (iv) the execution result and any OS-level command errors. These summaries are persisted to SQLite and exposed through authenticated dashboard API endpoints, enabling administrators to audit every autonomous decision. The dashboard renders these summaries as human-readable alert entries with color-coded severity levels (green/amber/red), ensuring that the system's autonomy does not come at the cost of administrative oversight—a key requirement for deployment in compliance-regulated environments.

VI. EXPECTED OUTPUT AND RESULTS

1. Detection and Mitigation Performance

The DQN agent was evaluated across 10,000 simulated attack events in the OpenAI Gym environment, spanning brute force, DDoS, SQL injection, port scanning, and multi-username probing categories with configurable attack intensity. The agent achieved a threat detection accuracy of 94.2% with a false positive rate of 3.1% and a missed detection rate of 2.7%. The successful

mitigation rate—defined as the correct action selected and successfully executed at the OS level—was 89.7%. The self-healing recovery module demonstrated a 97.3% successful rollback rate across 500 component compromise scenarios. Under 1000 events/minute stress load sustained for 30 minutes, the DQN inference engine maintained a mean response time of 23 ms (maximum: 41 ms), and the FastAPI backend sustained approximately 350 requests/second without event dropping or data loss. Dashboard updates remained consistent at 5-second intervals throughout the stress test duration.

Table 2: Performance Comparison — Proposed DQN System vs. Baselines

Metric	Proposed DQN System	Static Rule-Based	Threshold Anomaly
Detection Accuracy	94.2%	71.4%	78.9%
Mitigation Success Rate	89.7%	65.2%	—
False Positive Rate	3.1%	18.7%	22.4%
Missed Detection Rate	2.7%	28.6%	21.1%
Self-Healing Recovery	97.3%	N/A	N/A
Mean Inference Time	23 ms	~5 ms	~8 ms
Multi-Attack Coverage	Yes (4 types)	Partial	Limited

2. Agent Learning Behaviour

The DQN agent's training curve demonstrated consistent reward improvement over 2,000 training episodes. In early episodes (0–300), the agent exhibited near-random action selection with high false positive rates as the epsilon-greedy policy explored the action space. A significant reward inflection was observed around episode 500, corresponding to the agent learning early IP blocking before threshold breach—a proactive strategy not explicitly encoded in the reward function but emergent from Q-value optimization. By episode 1,500, the agent had converged to a stable policy achieving reward values within 5% of the maximum observed reward, with the training curve plateauing through episodes 1,500–2,000. The agent's discovered policy exhibited three non-obvious strategies: (i) proactive rate limiting before DDoS traffic reached saturation thresholds; (ii) honeypot redirection as a reconnaissance counter-measure that simultaneously gathered attacker intelligence; and (iii) selective temporary blocking followed by extended monitoring rather than immediate permanent blocking, reducing false positive rates for borderline events.

3. Operational Benefits

The system delivers several operational advantages beyond the quantitative metrics above. The transparent decision logging mechanism addresses the interpretability gap in RL-based security research, enabling administrators to maintain meaningful oversight. The self-healing capability substantially reduces MTTR—the primary operational KPI for security incident management—by automating the recovery phase that typically requires hours of manual administrator effort. The open-source Python-based implementation, with no commercial licensing dependencies, makes the system practically accessible to academic institutions and SMEs. The modular service-oriented architecture allows individual components to be upgraded, replaced, or scaled horizontally without system-wide redesign—supporting incremental migration into existing SOC workflows.

Future Scope

The current single-agent DQN architecture will be extended to a Multi-Agent Reinforcement Learning (MARL) framework, where cooperative agents monitor distinct network segments and share threat intelligence through a shared experience replay buffer—particularly suited to large enterprise and cloud-distributed environments where a single agent's state space becomes computationally intractable. Federated reinforcement learning will enable collaborative model training across organizational boundaries without sharing raw

network data, preserving data privacy while broadening agent generalizability across diverse network topologies and attack distributions.

Cloud-native deployment via Docker and Kubernetes on AWS, Azure, or Google Cloud Platform will enable horizontal auto-scaling of the FastAPI ingestion and DQN inference layers, with serverless execution of mitigation actions via cloud-native security APIs. Integration with SIEM platforms (Splunk, IBM QRadar, ELK Stack) and SOAR orchestration will close the loop between AI-driven detection and enterprise-grade incident response playbooks. Mapping the agent's action space and threat classification taxonomy to the MITRE ATT&CK framework will improve interoperability with commercial threat intelligence platforms and enable more precise analyst communication about detected attack campaigns. Integration of SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) will provide quantitative feature attribution for each agent decision, strengthening auditability for regulated environments and advancing the explainability of autonomous security systems. The architecture will also be extended to support Operational Technology (OT) and Industrial Control System (ICS) environments, where specialized state representations and action spaces for SCADA and Modbus protocol networks will be developed.

VII. CONCLUSION

This paper presented a Self-Healing Cyber Defence System Using Reinforcement Learning that addresses fundamental limitations of static, rule-based security architectures: the inability to adapt to novel threats, the absence of autonomous system recovery, and the opacity of automated decision logic. The proposed system integrates a DQN RL agent with real-time network event ingestion, a state representation engine, automated OS-level mitigation execution, a two-phase commit self-healing recovery module, persistent SQLite audit logging, and a real-time administrative security dashboard into a cohesive, transparent, production-oriented security ecosystem.

Evaluation results—94.2% detection accuracy, 89.7% mitigation success rate, and 97.3% self-healing recovery rate across 10,500 simulated events—demonstrate that RL-based autonomous defence substantially outperforms static baselines across all measured performance dimensions, while maintaining a 23 ms mean inference latency suitable for real-time deployment. The agent's emergent defensive strategies, including proactive rate limiting and honeypot-based intelligence gathering, demonstrate that RL can discover non-obvious policies that exceed the performance of expert-designed rule sets.

The transparent decision logging mechanism, which accompanies every agent action with a structured human-readable explanation and full Q-value trace, directly advances the auditability of AI-driven security systems and makes the proposed architecture practically deployable in compliance-sensitive enterprise environments. Academically, this work bridges theoretical RL concepts—Q-learning, experience replay, target network stabilization, and epsilon-greedy exploration—with real-world security engineering requirements, and establishes a strong, extensible foundation for future research in autonomous, federated, and cloud-native cyber defence.

Acknowledgment

The authors sincerely thank Dr. Pratap Singh (Paper Guide) for his invaluable guidance and supervision, Dr. Amit Kumar (Project Coordinator) for unwavering encouragement throughout this research, and Mr. Chunnu Lal (Program Officer) for his assistance and constructive insights. We are deeply grateful to Dr. Satender Kumar (Head of Department and Dean of Academics) and Dr. Brij Mohan Singh (Director, Quantum School of Technology) for providing the laboratory resources, computing infrastructure, and academic environment that enabled the successful completion of this work. We also acknowledge the open-source communities behind Python, TensorFlow, OpenAI Gym, and FastAPI, whose tools form the technical foundation of this system.

REFERENCES

1. A. O. Adebayo, M. A. Al-Habib, and S. M. Alzahrani, "A survey of brute force attack detection techniques in authentication systems," IJCNC, vol. 12, no. 4, pp. 45–60, July 2020.
2. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2018.
3. V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.
4. M. Standen, M. Lucas, D. Bowman, T. J. Richer, J. Kim, and D. Marriott, "CybORG: A Gym for the Development of Autonomous Cyber Agents," Proc. IJCAI-21 Workshop on Adaptive Cyber Defense, 2021.
5. N. Provos and D. Mazieres, "A future-adaptable password scheme," in Proc. USENIX Annual Technical Conf., San Diego, CA, USA, 1999, pp. 81–92.
6. ENISA, "Threat Landscape Report: Credential Stuffing and Brute Force Attacks," EU Agency for Cybersecurity, Tech. Rep., 2022.
7. R. Elderman, L. Pater, A. Thie, M. Drugan, and M. Wiering, "Adversarial Reinforcement Learning in a Cyber Security Simulation," in Proc. ICAART, 2017.
8. K. Malialis and D. Kudenko, "Multiagent router throttling: Decentralized coordinated response against DDoS attacks," in Proc. AAAI Workshop on AI for Cyber Security, 2015.
9. Microsoft Research, "CyberBattleSim," GitHub, 2021. [Online]. Available: https://github.com/microsoft/CyberBattleSim
10. M. Nasir, S. R. Islam, S. Noor, and A. Symington, "Autonomous intrusion response system using Q-learning," IEEE Access, vol. 9, pp. 120960–120978, 2021.
11. A. Barredo Arrieta et al., "Explainable AI (XAI): Concepts, taxonomies, opportunities and challenges," Information Fusion, vol. 58, pp. 82–115, June 2020.
12. F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.
13. S. T. Ribeiro, R. Singh, and C. Guestrin, "Why should I trust you?," in Proc. 22nd ACM SIGKDD, San Francisco, CA, 2016, pp. 1135–1144.
14. MITRE Corporation, "MITRE ATT&CK Framework," 2023. [Online]. Available: https://attack.mitre.org
15. FastAPI Documentation. [Online]. Available: https://fastapi.tiangolo.com
16. OpenAI Gym Documentation. [Online]. Available: https://www.gymlibrary.dev
17. Verizon, "Data Breach Investigations Report (DBIR)," 2024. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/
18. NIST, "Digital Identity Guidelines (SP 800-63)," National Institute of Standards and Technology, USA.
19. Shaiful Mahmud, Khaleel Khan Mohammed, Vasu Raj Jain, Sarthak Anandkumar Shah. "Artificial Intelligence-Driven Predictive Models for Identifying Risk Factors of Chronic Diseases"