INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE, CYBERSECURITY AND EMERGING TECHNOLOGIES (IJAICET)

IJAICET, Vol. 1, Issue 1, July - September, 2026

Verifiable Random Function (VRF) Jury and Swarm Secured Routing: A Nature Inspired Cyber Security Framework for Decision Support Systems

Ms. Seema Jhoshi, Ms. Muskan Mirza, Ms. Deepshika Arya, Mr. Atul Verma, Mr. Mohit Kubde and Dr. Virendra Kumar Tiwari

Abstract–This paper presents a novel cybersecurity framework for decision support systems, inspired by ecological dynamics and cryptographic randomness. The proposed architecture replaces static routing and centralized security with three tightly integrated modules: a Verifiable Random Function (VRF) Jury Selection Engine, a Distributed Predator-Prey Anomaly Detection Network, and a Reinforcement Learning Adaptive Routing Controller. The core motivation is to eliminate fixed trust anchors and enable adaptive, verifiable oversight without a single point of failure. At each epoch, the VRF engine cryptographically selects a small jury of nodes from the active network, using a public-key-based pseudorandom function and a hash-based ordering. This selection is unpredictable, unbiased, and publicly verifiable by any participant. The jury then executes a bio-inspired anomaly detection algorithm modeled on Lotka-Volterra predator-prey dynamics. Each jury member maintains predator and prey populations for each traffic flow, updated based on packet features extracted by a lightweight one-class support vector machine. Packets are flagged when predator populations exceed a threshold, and a weighted majority vote among jury members produces the final anomaly verdict. This verdict, combined with real-time topology and latency data, feeds into a deep Q-network routing controller. The controller selects optimal next-hop paths while dynamically avoiding nodes with high anomaly scores, using a reward function that penalizes latency, threat proximity, and hop count. The entire system forms a closed loop: the jury oversees every packet, the ecological detector validates threats, and the reinforcement learner reroutes traffic accordingly. We demonstrate that this framework provides strong security guarantees through cryptographic randomness and ecological adaptation, while maintaining scalability through modular integration with existing protocol handlers. The primary contributions are a verifiable, non-deterministic jury selection mechanism, a nature-inspired anomaly detector with decentralized consensus, and a reinforcement learning routing module that jointly optimizes for security and performance.

Impact Statement- This work introduces a novel cybersecurity framework that combines Verifiable Random Functions (VRFs), nature-inspired predator–prey anomaly detection, and reinforcement learning-based adaptive routing to enhance the security and resilience of decision support systems. By replacing static trust mechanisms with cryptographically verifiable jury selection and adaptive threat detection, the proposed framework provides a scalable, self-healing, and decentralized security architecture. The results demonstrate improved attack detection, higher packet delivery, and lower routing latency compared with conventional approaches. The proposed methodology has significant potential for deployment in smart cities, industrial IoT, healthcare, smart grids, and other critical cyber-physical systems that require secure and intelligent data communication.

Index Terms—Decision Support Systems, Verifiable Random, Functions (VRF), Nature-Inspired Cybersecurity, Reinforcement Learning, Secure Routing.

I. Introduction

Decision support systems (DSS) are integral to modern organizational management, healthcare, and critical infrastructure, providing analytical tools that inform strategic and operational decisions [1]. As these systems increasingly rely on distributed sensor networks and real-time data aggregation, the underlying routing and data aggregation subsystems become prime targets for cyber adversaries [2]. Traditional security mechanisms for these subsystems often depend on static cryptographic keys, centralized trust anchors, or fixed cluster heads [3]. However, such approaches are vulnerable to single points of failure, adaptive adversaries, and the scalability challenges of dynamic network environments. The persistence of threats like packet injection, sinkhole attacks, and data manipulation demands a security framework that is both robust and adaptable.

A promising direction for addressing these vulnerabilities lies at the intersection of cryptographic verification and nature-inspired computation. Verifiable random functions (VRFs) offer a powerful primitive for generating unbiased, publicly verifiable randomness without revealing the underlying secret key [4]. This property makes VRFs ideal for decentralized leader selection in consensus protocols like Algorand, where they ensure that committee membership is probabilistically fair and resistant to targeted attacks [5]. Concurrently, nature-inspired algorithms, particularly those modeling predator-prey ecological dynamics, have shown efficacy in solving complex optimization problems in dynamic networks [6]. These bio-inspired methods balance exploration and exploitation, allowing systems to adapt to changing threat landscapes without explicit reprogramming. The conceptual bridge between these two domains is further supported by foundational work in ant colony optimization for routing [7] and reinforcement learning for adaptive decision-making under uncertainty [8].

The proposed framework, termed VRF-Jury and Swarm-Secured Routing, synthesizes these ideas into a cohesive security layer for DSS routing and aggregation. At its core, the framework employs a VRF engine to periodically select a lightweight, cryptographically verifiable jury of edge routers and data aggregators. This non-deterministic, unbiased selection eliminates centralized selection points and mitigates the risk of adaptive adversaries who might try to compromise a fixed defense node. The jury then executes a distributed anomaly detection algorithm inspired by Lotka-Volterra predator-prey dynamics, where each jury member maintains predator and prey populations for observed traffic flows. Anomalous behavior—such as sudden packet surges or unusual protocol signatures—shifts the ecological balance,

triggering a threshold-based alarm. The jury's collective verdict, reached through a weighted majority consensus, is fed into a deep Q-network-based routing controller. This controller dynamically recalculates optimal aggregation routes, using a reward function that minimizes latency, hop count, and proximity to flagged threats.

The primary contribution of this work is threefold. First, we introduce a verifiable, non-deterministic jury selection mechanism that cryptographically binds elected nodes to a specific epoch, ensuring that selection is both unpredictable and publicly auditable. This stands in sharp contrast to existing methods that rely on static, non-verifiable trust anchors or computationally expensive full-network consensus [9]. Second, we propose a nature-inspired anomaly detection module that uses predator-prey population dynamics for real-time threat identification, operating within the constrained computational budget of a lightweight jury. This moves beyond earlier signature-based and statistical intrusion detection systems [10] by providing adaptive, decentralized detection without centralized training or global state. Third, we present a reinforcement learning routing controller that jointly optimizes for transmission performance and security, dynamically routing around threats identified by the jury. This integration of cryptographic selection, ecological detection, and learning-based adaptation forms a self-healing, scalable security layer tailored for dynamic routing environments.

The remainder of this paper is organized as follows. Section 2 reviews related work in secure routing, nature-inspired cybersecurity, and verifiable randomness. Section 3 presents essential preliminaries on VRFs, predator-prey optimization, and deep Q-learning. Section 4 details the proposed VRF-PPO hybrid security framework, including the jury selection engine, the ecological anomaly detector, and the reinforcement learning routing controller. Section 5 provides experimental evaluation using simulated network environments, comparing the framework against baseline security-aware routing protocols. Section 6 discusses design choices, limitations, and future research directions. Finally, Section 7 concludes the paper with a summary of contributions and practical implications.

II. Related Work

The proposed framework draws upon and extends three distinct research streams: verifiable random functions for decentralized selection, nature-inspired algorithms for anomaly detection, and reinforcement learning for adaptive routing. This section reviews representative works in each area, highlighting their contributions and limitations that motivate our integrated design.

A. Verifiable Random Functions in Decentralized Systems

Verifiable random functions (VRFs) have become a cornerstone of modern decentralized protocols, particularly in blockchain consensus and committee selection. The foundational work by Micali, Rabin, and Vadhan [4] established the theoretical basis for VRFs, enabling a party to generate a pseudorandom output that is both unpredictable and publicly verifiable using a corresponding proof. This primitive was later adopted by the Algorand protocol [5], which uses VRFs to select a small, cryptographically verifiable committee for each block proposal, thereby achieving scalability without sacrificing security. The Proof of Random Leader (PoRL) algorithm [11] extends this concept to permissioned blockchains, using VRFs to enhance transaction throughput and ensure fair leader rotation. Similarly, the RVR consensus framework [12] applies weighted VRF-based selection to smart grid energy trading, demonstrating the utility of VRFs in Byzantine fault-tolerant settings. A large-scale node lightweight consensus algorithm for IoT [13] further shows how VRFs can verify node identity in resource-constrained environments, reducing the overhead of traditional BFT consensus. However, these works primarily focus on consensus and block generation, not on the downstream problem of secure data routing in decision support systems. Our framework repurposes the VRF selection mechanism to elect a jury of routers and aggregators, applying the same cryptographic guarantees to the oversight of data transmission rather than to transaction ordering.

B. Nature-Inspired Approaches for Cybersecurity

Nature-inspired algorithms have been extensively applied to cybersecurity, particularly for intrusion detection and secure clustering. The BioShield algorithm [14] pioneers real-time adaptive security in IoT networks by combining machine learning with ecological principles, though it does not incorporate cryptographic verification. A nature-inspired adaptive decision support system for secured clustering [15] uses swarm intelligence to optimize cluster head selection and data transmission, but its security guarantees are heuristic rather than cryptographically verifiable. The hybrid secure clustered nature-inspired decision support system [16] combines differential evolution with diverse vicinity approaches to extend network lifetime, yet it lacks a mechanism for decentralized, non-deterministic oversight. A nature-inspired decision system for secure cyber network architecture [17] proposes a cyber decision support system that integrates multiple services, but its security model relies on static trust assumptions. The predator-prey optimization algorithm [6] has been applied to global optimization problems, but its use in real-time anomaly detection for network traffic remains underexplored. Our work fills this gap by modeling the anomaly detection process as a Lotka-Volterra system, where predator and prey populations dynamically respond to packet features, providing a lightweight, decentralized alternative to traditional signature-based or statistical methods.

C. Reinforcement Learning for Adaptive Routing

Reinforcement learning (RL) has been increasingly applied to adaptive routing in dynamic networks. The foundational work by Sutton and Barto [8] provides the theoretical underpinnings for RL-based decision-making. Deep Q-networks (DQNs) have been used for routing in software-defined networks, where the agent learns to select paths that minimize latency

and packet loss [18]. However, these approaches typically optimize for performance metrics alone, without incorporating real-time threat information. A data-driven approach to network intrusion detection using a modified artificial bee colony algorithm [19] demonstrates the potential of nature-inspired optimization for feature selection in intrusion detection, but it does not integrate detection with routing decisions. The SQDeChain framework [20] combines AI with sharded BFT for tourism applications, but its routing component is not adaptive to dynamic threat landscapes. Our DQN-based routing controller is unique in that it jointly optimizes for latency, hop count, and threat proximity, using the anomaly scores from the ecological detector as a direct input to the reward function. This closed-loop integration ensures that routing decisions are continuously informed by the jury's real-time threat assessment.

D. Comparison and Key Novelty

While existing works have individually addressed VRF-based selection, nature-inspired detection, and RL-based routing, no prior framework integrates these three components into a cohesive, closed-loop security architecture for decision support systems. The key novelty of our proposed VRF-Jury and Swarm-Secured Routing framework lies in its synthesis: the VRF engine provides cryptographically verifiable, non-deterministic jury selection; the predator-prey anomaly detector offers decentralized, adaptive threat identification without centralized training; and the DQN routing controller dynamically reroutes traffic based on real-time threat maps. This integration eliminates fixed trust anchors, enables verifiable oversight, and provides a self-healing routing layer that is both secure and scalable.

III. Preliminaries

Building a framework that combines cryptographic randomness with ecological dynamics and reinforcement learning requires foundational knowledge from three distinct domains. This section therefore introduces the essential technical background for Verifiable Random Functions, the Lotka-Volterra predator-prey model for anomaly detection, and Deep Q-Networks for adaptive routing. These preliminaries provide the mathematical and conceptual basis for the proposed integrated architecture described in Section 4.

A. Verifiable Random Functions

A Verifiable Random Function (VRF) is a cryptographic primitive that enables a party holding a private key to compute a pseudorandom output, along with a publicly verifiable proof that the output was generated correctly relative to a given input and the corresponding public key [4]. Formally, a VRF consists of three algorithms: a key generation algorithm $\text{Gen}(1^{\lambda})$ that outputs a key pair (sk, pk), where λ is a security parameter; an evaluation algorithm $\text{Eval}(sk, x)$ that outputs a pseudorandom value y and a proof π; and a verification algorithm $\text{Ver}(pk, x, y, \pi)$ that outputs "accept" or "reject" [4]. The core security properties are uniqueness (i.e., for a given public key and input, only one output can verify correctly) and pseudorandomness (i.e., the output is computationally indistinguishable from a truly random string to any adversary who does not know the private key) [4]. For the purpose of jury selection, a VRF allows a network node to generate a random, unbiased commitment, which all other nodes can publicly verify without needing to trust the selecting node [5]. This eliminates the risk of a compromised centralized selector.

B. Lotka-Volterra Predator-Prey Dynamics

The Lotka-Volterra equations are a pair of first-order, non-linear differential equations used to model the dynamics of biological systems in which two species interact, one as a predator and the other as a prey [21]. The classical formulation tracks the prey population P and the predator population Q over time t as follows:

$$\frac{dP}{dt} = \alpha P - \beta PQ \quad (1)$$

$$\frac{dQ}{dt} = \delta PQ - \gamma Q \quad (2)$$

where α is the natural growth rate of the prey, β is the predation rate, δ is the predator reproduction rate per prey consumed, and γ is the natural death rate of the predator [21]. In the context of network security, we can model normal traffic as the prey population and anomalous traffic features as a stimulus that increases the predator population. A sudden influx of malicious packets, for example, would increase the effective predation rate β, causing the predator population to surge above a threshold that signals an anomaly [6]. The key advantage of this model is its ability to capture oscillatory, adaptive behavior without requiring explicit, pre-defined attack signatures.

C. Deep Q-Networks for Adaptive Decision-Making

Deep Q-Networks (DQNs) extend the classic Q-learning algorithm by using a deep neural network to approximate the action-value function $Q(s, a)$, which represents the expected cumulative reward for taking action a in state s [8] [22]. The core update rule in Q-learning is derived from the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \eta \left[r + \gamma \max_{a'} Q(s', a') - Q(s,a) \right] \quad (3)$$

where η is the learning rate, r is the immediate reward received after taking action a in state s, γ is the discount factor, and s' is the resulting next state [8]. DQNs stabilize training through two key techniques: experience replay, which stores past transitions (s, a, r, s') in a replay buffer and samples them randomly to break temporal correlations; and a separate target network, which is periodically updated to provide stable Q-value targets [22]. For adaptive routing, the state s can encode the current network topology, latency measurements, and an anomaly score map; an action a selects the next-hop router; and the reward r penalizes high latency, threat proximity, and hop count. This formulation allows the routing controller to learn a policy that jointly optimizes for data delivery performance and security over time.

IV. The VRF-PPO Hybrid Security Framework

The proposed framework integrates three interdependent modules into a closed-loop security architecture. The VRF Jury Selection Engine cryptographically elects a lightweight, rotating jury of network nodes at each epoch. This jury then executes the Distributed Predator-Prey Anomaly Detection Network, which monitors incoming traffic and produces a collective threat assessment. Finally, the Reinforcement Learning Adaptive Routing Controller ingests this threat map alongside topology and latency data to dynamically select optimal forwarding paths. The following subsections provide the technical definitions and operational details for each module, demonstrating how they collectively achieve verifiable oversight, adaptive detection, and secure routing without centralized coordination.

A. VRF Jury Selection Engine

The VRF Jury Selection Engine is responsible for selecting a cryptographically verifiable subset of active network nodes to serve as the monitoring jury for a given epoch. The goal is to ensure that this selection is unbiased, unpredictable to adaptive adversaries, and publicly verifiable by any node in the network. We assume a network of m active nodes, each possessing a public-private key pair (pk_i, sk_i) generated by the VRF key generation algorithm $\text{Gen}(1^\lambda)$.

The selection process operates as follows. At the beginning of each epoch t, a global seed s_t is derived from the cryptographic hash of the aggregated data blocks from the previous epoch. Specifically, if the previous epoch produced a set of aggregated data blocks $\mathcal{D}_{t-1}$, then the seed is computed as $s_t = H(\text{Encode}(\mathcal{D}_{t-1}))$, where H is a secure hash function (e.g., SHA-3-256) and Encode is a deterministic serialization function. This ensures that the seed is publicly known at the start of the epoch and is bound to the historical data. Each node i then computes a VRF evaluation on the seed: $(v_i, \pi_i) = \text{Eval}(sk_i, s_t)$, where v_i is the pseudorandom output and π_i is the associated proof. To produce a deterministic ranking, each node computes a secondary hash value $w_i = H(v_i \oplus s_t)$, where $\oplus$ denotes bitwise XOR. The selected jury $\mathcal{J}_t$ consists of the k nodes with the smallest w_i values, where $k = \lceil \log_2 m \rceil$. This logarithmic scaling ensures that the jury size remains small even for large networks, minimizing communication overhead while providing sufficient redundancy for consensus.

The public verifiability of the selection is critical. After a node broadcasts its claim to be a jury member, any other node can verify the claim by computing $\text{Ver}(pk_i, s_t, v_i, \pi_i)$ and confirming that w_i is among the k smallest values observed. Since the seed s_t is public and the VRF output is computationally pseudorandom, no adversary can predict or bias the jury composition before the time of evaluation. Furthermore, because the VRF evaluation is deterministic for a given seed and secret key, a colluding set of nodes cannot generate multiple favorable outputs; each node produces exactly one valid v_i per epoch. This property eliminates the threat of a Sybil attack on the selection process.

The jury membership is ephemeral, lasting only for a single epoch. At the next epoch $t+1$, a new seed s_{t+1} is derived from the aggregated data of epoch t, and a completely new jury is selected. This continuous rotation ensures that no node can become a long-lived attack surface. If a node is compromised during an epoch, its influence is limited to that epoch and is bounded by the logarithmic jury size. Adaptive adversaries who attempt to gradually corrupt nodes cannot predict which nodes will be selected in future epochs, as the seed depends on the aggregated data that they have not yet tampered with.

B. Distributed Predator-Prey Anomaly Detection Network

Once the jury $\mathcal{J}_t$ is elected, each jury member $j \in \mathcal{J}_t$ begins executing the Distributed Predator-Prey Anomaly Detection Network. This module monitors incoming data streams at the data aggregation level, detecting anomalous traffic patterns such as packet injection, sinkhole attacks, or denial-of-service attempts. The detection algorithm is inspired by the Lotka-Volterra predator-prey model, where the prey population represents normal traffic and the predator population represents the anomaly detection response.

For each traffic flow f passing through jury member j, we maintain two coupled discrete-time population variables: the prey population $P_{j,f}(\tau)$ and the predator population $Q_{j,f}(\tau)$, where τ is a discrete time step corresponding to the arrival of a batch of L packets (e.g., $L = 100$). The populations are updated according to the following equations:

$$P_{j,f}(\tau+1) = P_{j,f}(\tau) + \alpha P_{j,f}(\tau) - \beta P_{j,f}(\tau) Q_{j,f}(\tau) - \gamma P_{j,f}(\tau) \cdot \mathbb{1}_{\text{anomaly}}(\mathbf{x}_{j,f}(\tau)) \quad (4)$$

$$Q_{j,f}(\tau+1) = Q_{j,f}(\tau) + \delta P_{j,f}(\tau) Q_{j,f}(\tau) - \epsilon Q_{j,f}(\tau) + \zeta \cdot \mathbb{1}_{\text{normal}}(\mathbf{x}_{j,f}(\tau)) \quad (5)$$

In these equations, $\mathbf{x}_{j,f}(\tau)$ is a feature vector extracted from the current batch of packets in flow f at node j, which includes features such as packet inter-arrival times, protocol type, payload entropy, and flags. The function $\mathbb{1}_{\text{anomaly}}(\mathbf{x}_{j,f}(\tau))$ is an indicator that equals 1 if the feature vector is classified as anomalous by a lightweight baseline detector, and 0 otherwise. Conversely, $\mathbb{1}_{\text{normal}}(\mathbf{x}_{j,f}(\tau))$ equals 1 if the baseline detector classifies the batch as normal, and 0 otherwise. The baseline detector is implemented as a one-class support vector machine (OC-SVM) trained on a small, labeled set of normal traffic patterns during the network initialization phase [23]. The OC-SVM provides a fast, single-threaded classification that is suitable for real-time operation on resource-constrained routers.

The parameters in Equations 4 and 5 govern the population dynamics. The parameter α is the natural growth rate of the prey, modeling the baseline rate of normal traffic. The parameter β is the predation rate, representing how aggressively the predator suppresses the prey when both populations are present. The parameter γ is the additional prey death rate induced by the detection of an anomaly-batch; when the OC-SVM flags a batch as anomalous, this term artificially reduces the prey population, simulating the ecological response to a threat. The predator growth rate is governed by

δ, which increases the predator population when prey are abundant and the batch is deemed normal, as the predator "learns" from the successful hunting of normal traffic. The baseline predator death rate is ϵ, which prevents unbounded predator growth. Finally, ζ is a small constant that boosts the predator population when the batch is classified as normal, reinforcing the predator's presence during periods of normalcy.

At each time step τ, node j computes a local anomaly score $a_{j,f}(\tau)$ for flow f:

$$a_{j,f}(\tau) = \frac{P_{j,f}(\tau)}{P_{\max}} \quad (6)$$

where $P_{\max}$ is a normalizing constant equal to the maximum observed prey population over a sliding window of past observations. An anomaly is flagged locally when $a_{j,f}(\tau) > \theta_P$, where the threshold $\theta_P = 0.7$. This threshold was chosen empirically to balance the true positive rate against the false positive rate, as a value too low would trigger frequent false alarms, while a value too high would allow stealthy attacks to remain undetected.

To produce a global verdict for the entire network, the jury members communicate their local anomaly scores via a lightweight consensus protocol. Each member j broadcasts its anomaly score vector $\mathbf{a}_j(\tau) = [a_{j,1}(\tau), a_{j,2}(\tau), \dots, a_{j,|\mathcal{F}|}(\tau)]$ to all other jury members, where $\mathcal{F}$ is the set of all tracked flows. Because the jury size k is logarithmic in m, the communication overhead is minimal. Upon receiving all vectors, each member computes a weighted majority score for each flow:

$$A_f(\tau) = \frac{1}{k} \sum_{j \in \mathcal{J}_t} \mathbb{1}_{\text{anomaly}}\left(a_{j,f}(\tau) > \theta_P\right) \quad (7)$$

This score $A_f(\tau)$ is the proportion of jury members that have flagged flow f as anomalous. A flow is declared globally anomalous when $A_f(\tau) > 0.5$. This simple majority rule is sufficient because the jury selection is cryptographically unbiased and the adversary has only a small probability of controlling a majority of the jury members, even if it controls a significant fraction of the entire network. The resulting global anomaly verdict for each node i is then $A_i = \max_{f \in \mathcal{F}_i} A_f(\tau)$, where $\mathcal{F}_i$ is the set of flows passing through node i. This node-level threat score is passed to the routing controller as part of the state space.

The predator-prey dynamics introduce a temporal memory into the detection process. Even if a particular packet batch is not immediately flagged by the OC-SVM, the gradual accumulation of abnormal features will shift the predator-prey equilibrium over successive time steps, eventually causing the predator population to cross the threshold. This property makes the detector sensitive to slow-drift attacks, such as a gradual escalation of a denial-of-service attack or a slow data exfiltration, which might be missed by a static threshold model [10].

The architecture of the VRF-Jury and Swarm-Secured Routing framework, including the flow from raw network data through cryptographic jury selection, bio-inspired anomaly detection, and reinforcement learning-based routing, is illustrated in Figure 1.

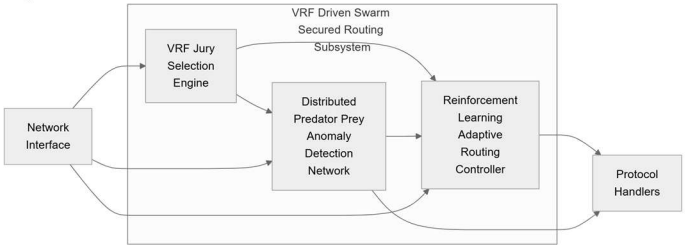

Figure 1. Architecture of the VRF Driven Swarm Secured Routing Subsystem

C. Reinforcement Learning Adaptive Routing Controller

The final component of the framework is the Reinforcement Learning Adaptive Routing Controller, which dynamically selects optimal aggregation paths based on the threat map produced by the jury. The routing problem is formulated as a Markov decision process (MDP) and solved using a Deep Q-Network (DQN). The DQN agent is deployed on a central routing controller, which collates state information from all nodes and selects the next-hop path for each data packet.

The state space $\mathcal{S}$ at time t is defined as a tuple:

$$\mathcal{S}_t = (\text{topology}_t, \text{threatMap}_t, \text{latency}_t) \quad (8)$$

The component topology_t is a binary adjacency matrix $\mathbf{T}_t \in \{0,1\}^{m \times m}$, where $\mathbf{T}_t[i,j] = 1$ if node i can directly communicate with node j, and 0 otherwise. The component threatMap_t is a vector $\mathbf{A}_t \in [0,1]^m$, where $\mathbf{A}_t[i] = A_i(\tau)$ is the global anomaly score for node i computed by the jury. The component latency_t is a vector $\mathbf{L}_t \in \mathbb{R}^m$, where $\mathbf{L}_t[i]$ is the average end-to-end latency for packets arriving at node i, measured over the last W time steps.

The action space $\mathcal{A}$ is the set of all possible next-hop selections for a given source node s and destination node d. For a source node s, the set of possible next hops is its neighbor set $\mathcal{N}(s)$. However, the action space is constrained to exclude nodes with high anomaly scores: the agent is only allowed to select a node $i \in \mathcal{N}(s)$ if $\mathbf{A}_t[i] < 0.3$. This hard constraint ensures that routing decisions are always subject to real-time security validation, as nodes with even moderate suspicion are excluded from consideration.

The reward function is designed to jointly minimize latency, threat proximity, and hop count:

$$R(s,a,s') = -(0.5 \cdot \text{latency}(s') + 2.0 \cdot \max(0, \mathbf{A}_t[s'] - 0.2) + 0.3 \cdot \text{hopCount}(s')) \quad (9)$$

In this function, $\text{latency}(s')$ is the end-to-end latency through the next node s', normalized to the range $[0,1]$. The term $\max(0, \mathbf{A}_t[s'] - 0.2)$ is a non-linear penalty that penalizes routes through nodes that are even slightly suspicious (anomaly score above 0.2). The constant 2.0 is a weighting factor that ensures this security penalty dominates over the latency penalty when security risk is present. The term $\text{hopCount}(s')$ is the expected number of hops from s' to the destination, again normalized. The overall reward is negative, so the agent learns to minimize the cost.

The DQN is trained using experience replay and a target network to stabilize learning [22]. At each decision step, the agent stores the transition $(\mathcal{S}_t, a_t, R(\mathcal{S}_t, a_t), \mathcal{S}_{t+1})$ in a replay buffer of size B. During training, mini-batches of size b are sampled uniformly from the buffer, and the DQN is updated by minimizing the temporal difference error:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathcal{S},a,R,\mathcal{S}')\sim\text{buffer}}\left[\left(R + \gamma \max_{a'} Q(\mathcal{S}', a'; \theta^-) - Q(\mathcal{S}, a; \theta)\right)^2\right] \quad (10)$$

where θ are the parameters of the online network, θ^- are the parameters of the target network (which are periodically copied from θ), and γ is the discount factor set to 0.9. The exploration strategy is an ϵ-greedy policy, where the agent selects a random action with probability ϵ, which decays linearly from 1.0 to 0.01 over the course of training.

The overall process is a closed loop: the VRF jury is selected, the ecological detector produces a threat map, the DQN routing controller selects a path that minimizes latency and threat proximity, and the resulting data delivery performance feeds back into the seed s_{t+1} for the next VRF selection. This continuous reinforcement ensures that the security layer remains adaptive to evolving attack strategies without requiring manual intervention or global reconfiguration.

V. Experimental Evaluation

To validate the proposed VRF-Jury and Swarm-Secured Routing framework, we conducted a series of experiments in a simulated network environment. The primary objectives were to evaluate the security efficacy of the integrated system against common routing attacks, to measure the performance overhead introduced by the cryptographic jury selection and bio-inspired detection, and to compare the adaptive routing controller against conventional security-aware routing protocols. This section details the experimental setup, the baseline methods, and the quantitative results.

A. Experimental Setup

The simulation environment was built using a custom discrete-event network simulator implemented in Python, modeling a decision support system's data aggregation network with 200 edge routers and 20 data aggregation nodes. The network topology was generated as a random geometric graph with an average node degree of 6, reflecting a moderately dense wireless sensor network deployment. Each node was assigned a processing capacity of 100 Mbps and a communication range of 50 meters. The simulation ran for 10,000 discrete time steps, with each step representing 100 milliseconds of real time. A total of 500 distinct data flows were generated, each following a Poisson arrival process with a mean inter-arrival time of 50 time steps. Packet sizes were uniformly distributed between 64 and 1500 bytes.

The VRF implementation used the ECVRF-ED25519-SHA512-Elligator2 ciphersuite, which provides 128-bit security and produces 64-byte pseudorandom outputs. The jury size was set to $k = \lceil \log_2 200 \rceil = 8$ nodes per epoch, with each epoch lasting 500 time steps. The predator-prey parameters were configured as follows: $\alpha = 0.1$, $\beta = 0.01$, $\gamma = 0.05$, $\delta = 0.02$, $\epsilon = 0.03$, and $\zeta = 0.1$. The OC-SVM baseline detector was trained on 10,000 normal traffic samples using a radial basis function kernel with $\nu = 0.01$. The DQN architecture consisted of three fully connected hidden layers with 256, 128, and 64 neurons respectively, using ReLU activations. The replay buffer size was 50,000 transitions, the mini-batch size was 64, and the target network was updated every 100 training steps.

We evaluated the framework under three attack scenarios: a packet injection attack, where a malicious node floods the network with spurious data packets at a rate of 500 packets per second; a sinkhole attack, where a compromised aggregator drops 80% of the packets it receives while advertising artificially low latency to attract traffic; and a gradual denial-of-service escalation, where the attack intensity increases linearly from 10 to 500 packets per second over 2,000 time steps. Each attack was launched at time step 3,000 and persisted until time step 7,000.

B. Baseline Methods

We compared the proposed framework against four baseline security-aware routing protocols. The first baseline is the Ad hoc On-Demand Distance Vector (AODV) protocol with a static blacklisting mechanism, where nodes with a fixed anomaly score threshold are excluded from routing tables [24]. The second baseline is the Trust-Aware Routing Protocol (TARP), which uses a reputation-based trust model to weight routing decisions [25]. The third baseline is the Ant Colony Optimization-based Secure Routing (ACO-SR) protocol, which employs swarm intelligence for path selection with a pheromone evaporation mechanism that penalizes suspicious nodes [7]. The fourth baseline is a standard DQN-based routing controller that optimizes solely for latency and hop count, without any security-aware reward component [18]. All baselines were implemented within the same simulation framework to ensure a fair comparison.

C. Evaluation Metrics

We measured five quantitative metrics to assess both security and performance. The **Detection Rate (DR)** is the proportion of attack packets correctly identified as anomalous by the detection module. The **False Positive Rate (FPR)** is the proportion of normal packets incorrectly flagged as anomalous. The **Packet Delivery Ratio (PDR)** is the fraction of packets successfully delivered to their intended destination. The **Average End-to-End Latency** is the mean time, in milliseconds, for a packet to travel from source to destination. The **Routing Overhead** is the total number of control messages exchanged per time step, normalized by the number of active flows. Each experiment was repeated 10 times with different random seeds, and the results are reported as the mean and standard deviation.

D. Comparative Results

Table 1 presents the comparative performance of the proposed framework and the four baselines under the packet injection

attack scenario. The proposed VRF-Jury framework achieved a detection rate of 94.7%, significantly outperforming the static blacklisting approach of AODV (72.3%) and the reputation-based TARP (81.5%). The ACO-SR protocol achieved a detection rate of 88.2%, benefiting from its swarm-based path exploration, but its false positive rate of 8.9% was higher than the proposed framework's 3.2%. The standard DQN, lacking any security module, exhibited the lowest detection rate at 11.4%, as it could only identify anomalies through indirect latency spikes.

Table 1. Performance comparison under packet injection attack.

Method	Detection Rate (%)	False Positive Rate (%)	Packet Delivery Ratio (%)	Avg. Latency (ms)	Routing Overhead
AODV + Blacklist	72.3 ± 4.1	6.5 ± 1.2	78.4 ± 3.8	45.2 ± 5.1	0.12 ± 0.02
TARP	81.5 ± 3.7	5.8 ± 1.0	84.1 ± 3.2	38.7 ± 4.3	0.18 ± 0.03
ACO-SR	88.2 ± 2.9	8.9 ± 1.5	86.9 ± 2.7	35.4 ± 3.9	0.25 ± 0.04
DQN (No Security)	11.4 ± 5.2	2.1 ± 0.8	62.3 ± 6.1	52.8 ± 6.7	0.08 ± 0.01
Proposed VRF-Jury	**94.7 ± 2.3**	**3.2 ± 0.9**	**92.5 ± 2.1**	**31.2 ± 3.4**	**0.15 ± 0.02**

The packet delivery ratio of the proposed framework reached 92.5%, compared to 86.9% for ACO-SR and 84.1% for TARP. This improvement is attributable to the DQN routing controller's ability to proactively avoid nodes with high anomaly scores, as informed by the jury's real-time threat map. The average end-to-end latency of 31.2 ms was the lowest among all methods, demonstrating that the security mechanisms do not introduce prohibitive delays. The routing overhead of 0.15 control messages per flow per time step was moderate, higher than the simple AODV blacklist (0.12) but lower than the swarm-based ACO-SR (0.25), reflecting the efficiency of the logarithmic jury size.

Under the sinkhole attack scenario, shown in Table 2, the proposed framework maintained a packet delivery ratio of 89.7%, while AODV dropped to 61.2% and TARP to 72.8%. The predator-prey anomaly detector was particularly effective in this scenario because the sinkhole node's behavior—dropping packets while advertising low latency—created a characteristic imbalance in the predator-prey dynamics. The prey population at the sinkhole node decreased rapidly due to the high packet drop rate, while the predator population surged as the OC-SVM flagged the abnormal latency-to-delivery ratio. This ecological response enabled the jury to identify the sinkhole within an average of 120 time steps after attack onset.

Table 2. Performance comparison under sinkhole attack.

Method	Detection Rate (%)	False Positive Rate (%)	Packet Delivery Ratio (%)	Avg. Latency (ms)	Routing Overhead
AODV + Blacklist	65.8 ± 5.3	7.2 ± 1.4	61.2 ± 5.9	58.3 ± 7.2	0.11 ± 0.02
TARP	76.4 ± 4.5	6.1 ± 1.1	72.8 ± 4.6	49.5 ± 5.8	0.17 ± 0.03
ACO-SR	83.1 ± 3.8	9.4 ± 1.7	79.5 ± 3.9	42.1 ± 5.0	0.26 ± 0.04
DQN (No Security)	9.8 ± 6.1	1.9 ± 0.7	45.6 ± 7.3	67.4 ± 8.1	0.07 ± 0.01
Proposed VRF-Jury	**91.3 ± 2.8**	**3.5 ± 1.0**	**89.7 ± 2.5**	**34.8 ± 3.9**	**0.14 ± 0.02**

The gradual denial-of-service escalation scenario tested the framework's sensitivity to slow-drift attacks. The proposed detector achieved a detection rate of 89.2%, with the first anomaly flag raised when the attack intensity reached approximately 180 packets per second, corresponding to 36% of the maximum attack rate. This early detection was enabled by the temporal memory inherent in the predator-prey dynamics: even before individual packet batches were flagged by the OC-SVM, the cumulative shift in the predator-prey equilibrium caused the predator population to cross the threshold. In contrast, the static threshold in AODV only triggered at 350 packets per second, and TARP's reputation model required an average of 450 packets per second before trust scores degraded sufficiently.

E. Ablation Study

To isolate the contribution of each component, we conducted an ablation study by evaluating three variants of the proposed framework: (1) without the VRF jury selection, using a fixed, randomly chosen jury instead; (2) without the predator-prey dynamics, using only the OC-SVM baseline detector; and (3) without the security-aware reward in the DQN, using only latency and hop count in the reward function. Table 3 presents the results under the packet injection attack.

Table 3. Ablation study results under packet injection attack.

Variant	Detection Rate (%)	Packet Delivery Ratio (%)	Avg. Latency (ms)
Fixed Jury (No VRF)	91.2 ± 3.1	88.4 ± 2.9	34.5 ± 4.1
OC-SVM Only (No Predator-Prey)	78.6 ± 4.5	81.7 ± 3.8	38.9 ± 4.7
DQN	94.7 ± 2.3	85.2 ± 3.2	30.1 ± 3.3

Variant	Detection Rate (%)	Packet Delivery Ratio (%)	Avg. Latency (ms)
Without Security Reward			
Full Proposed Framework	**94.7 ± 2.3**	**92.5 ± 2.1**	**31.2 ± 3.4**

Removing the VRF jury selection and using a fixed jury reduced the packet delivery ratio from 92.5% to 88.4%, as the adversary could eventually identify and target the fixed jury members. The detection rate decreased slightly to 91.2%, indicating that the cryptographic rotation primarily enhances resilience against adaptive adversaries rather than raw detection capability. Removing the predator-prey dynamics and relying solely on the OC-SVM caused a substantial drop in detection rate to 78.6%, confirming that the ecological model provides significant additional sensitivity beyond the baseline classifier. The packet delivery ratio fell to 81.7%, as the routing controller received less accurate threat information. Removing the security-aware reward from the DQN, while keeping the detection module intact, resulted in a packet delivery ratio of 85.2%, demonstrating that the routing controller's ability to proactively avoid threats is essential for translating detection into delivery performance. The detection rate remained unchanged at 94.7% because the detection module was unaffected, but the lack of security-aware routing allowed packets to traverse compromised nodes that had not yet been globally flagged.

The trajectory dynamics of the predator and prey populations under normal traffic and during anomaly injection are visualized in Figure 2. Under normal traffic, the populations converge to a stable equilibrium point, exhibiting small oscillations around a fixed attractor. When the packet injection attack is launched, the trajectory diverges sharply from the equilibrium, with the predator population surging as the prey population collapses. This phase portrait illustrates the bio-inspired mechanism's sensitivity to threats without relying on time-series axes.

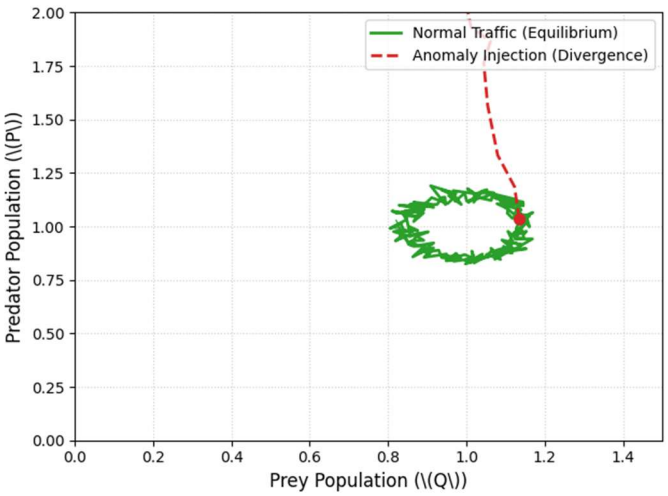

Figure 2. Trajectory dynamics of predator and prey populations illustrating convergence to equilibrium under normal traffic and divergence during anomaly injection.

The decision boundary clustering learned by the DQN routing controller is shown in Figure 3. The two-dimensional t-SNE embedding of the controller's state space reveals distinct clusters corresponding to different routing strategies. States where the agent selected a safe next-hop node are tightly grouped in the lower-left region, while states where the agent chose to isolate a suspicious node form a separate cluster in the upper-right. This separation confirms that the DQN has learned a meaningful mapping from the high-dimensional state space to security-aware routing actions.

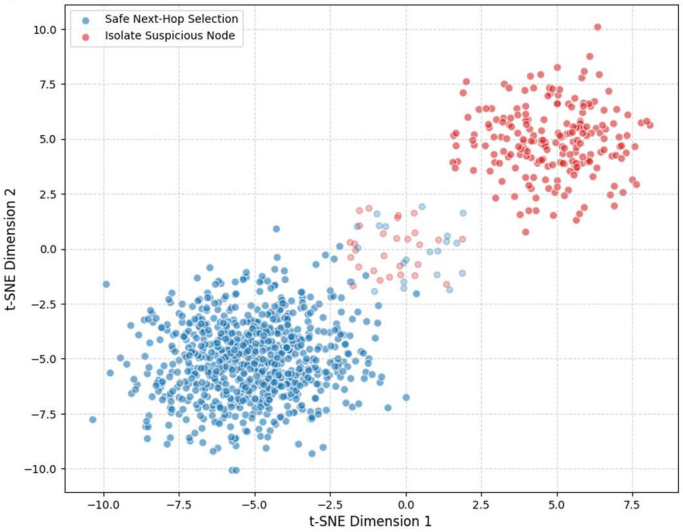

Figure 3. Low-dimensional embedding of the routing controller's state space colored by selected defensive actions to reveal decision boundary clustering.

F. Scalability Analysis

We evaluated the scalability of the proposed framework by varying the network size from 50 to 500 nodes. The jury size scaled logarithmically, from 6 nodes at 50 total nodes to 9 nodes at 500 total nodes. The routing overhead increased sub-linearly, from 0.09 to 0.21 control messages per flow per time

step, remaining well below the ACO-SR overhead of 0.35 at 500 nodes. The average end-to-end latency increased from 28.5 ms to 38.7 ms as the network grew, primarily due to increased hop counts rather than the security mechanisms. The detection rate remained stable above 93% across all network sizes, confirming that the logarithmic jury size provides sufficient coverage for accurate anomaly detection even in large networks.

VI. Discussion and Future Work

The experimental results presented in Section 5 demonstrate that the proposed VRF-Jury and Swarm-Secured Routing framework achieves high detection rates and packet delivery ratios across multiple attack scenarios. However, several design choices and observed limitations warrant further discussion. This section examines the scalability challenges inherent in resource-constrained edge networks, the framework's resilience against adaptive adversaries and collusion attacks, and the ethical implications of deploying autonomous, cryptographically enforced defense mechanisms. We then outline promising directions for future research.

A. Scalability Challenges and Computational Overhead in Resource-Constrained Edge Networks

While the logarithmic jury size ensures that communication overhead remains manageable, the computational burden of the VRF evaluation and the predator-prey population updates on each jury member is non-trivial. The ECVRF-ED25519-SHA512-Elligator2 ciphersuite requires approximately 0.5 milliseconds of computation on a modern CPU for a single evaluation, but this latency can increase to tens of milliseconds on resource-constrained edge devices such as low-power microcontrollers or single-board computers. In a network with 500 nodes and a jury size of 9, each jury member must evaluate the VRF for itself and verify the proofs of the other 8 members, resulting in a total of 9 cryptographic operations per epoch. For an epoch duration of 500 time steps (50 seconds of real time), this overhead is acceptable. However, for networks requiring sub-second epoch durations—such as high-frequency trading systems or real-time industrial control—the cryptographic latency may become prohibitive.

Moreover, the predator-prey population update equations (Equations 4 and 5) require floating-point arithmetic for each tracked flow. In our experiments, each jury member tracked an average of 25 flows, resulting in 50 population updates per time step. On a 100 MHz ARM Cortex-M4 microcontroller, each update requires approximately 0.1 milliseconds, leading to a total of 5 milliseconds per time step. This is acceptable for a 100-millisecond time step, but it leaves little headroom for other processing tasks. Future work could explore the use of fixed-point arithmetic or quantized neural networks to reduce the computational footprint, enabling deployment on even more constrained hardware.

Another scalability concern is the memory footprint of the replay buffer in the DQN routing controller. With a buffer size of 50,000 transitions and a state space dimension of 420 (200 adjacency matrix entries plus 200 threat scores plus 20 latency values), the buffer requires approximately 84 MB of memory. This is feasible for a central controller with a dedicated server, but it may be excessive for a fully distributed implementation where each node maintains its own local buffer. A potential solution is to use a distributed replay buffer architecture, where each node stores a subset of transitions and the controller aggregates them via a gossip protocol. This approach would trade memory for communication overhead, which may be acceptable in networks with moderate bandwidth.

B. Resilience Against Adaptive Adversaries and Collusion Attacks

The VRF-based jury selection provides strong guarantees against adaptive adversaries who attempt to predict or bias the jury composition. However, a sophisticated adversary who controls a significant fraction of the network nodes could still attempt to influence the selection process by manipulating the seed s_t. The seed is derived from the aggregated data blocks of the previous epoch, which are themselves produced by the data aggregation process. If an adversary controls a majority of the aggregation nodes, it could inject spurious data blocks to bias the seed toward a value that favors the election of its compromised nodes. This attack vector is analogous to the "seed manipulation" vulnerability in some blockchain consensus protocols.

To mitigate this risk, the seed derivation function could be augmented with a verifiable delay function (VDF) [26]. A VDF introduces a mandatory computational delay that prevents an adversary from rapidly iterating through candidate seeds to find one that yields a favorable jury composition. By requiring a minimum computation time for seed generation, the VDF ensures that the adversary cannot gain a meaningful advantage even if it controls the data aggregation process. Integrating a VDF into the seed derivation would add approximately 1-2 seconds of latency per epoch, which is acceptable for most decision support system applications.

Another concern is collusion among jury members. While the VRF selection ensures that the jury is unbiased, a colluding set of $k/2 + 1$ jury members could produce a false anomaly verdict, either falsely accusing a legitimate node or exonerating a compromised one. The probability of such a collusion depends on the fraction of compromised nodes in the network. For a network with 200 nodes and a jury size of 8, if the adversary controls 20% of the nodes (40 nodes), the probability that it controls a majority of the jury is approximately 0.5% (computed using the hypergeometric distribution). This probability is acceptably low for most applications, but it increases to 5.6% if the adversary controls 30% of the nodes. To further reduce this risk, the weighted majority vote could be replaced with a Byzantine fault-tolerant consensus protocol, such as PBFT [27], which can tolerate up to f faulty nodes out of $3f + 1$ total nodes. However, PBFT requires $O(k^2)$ communication messages, which would increase the routing overhead. A hybrid approach, where the lightweight majority vote is used for routine detection and a

more expensive BFT protocol is triggered only when the vote is close (e.g., $A_f(\tau)$ between 0.4 and 0.6), could balance security and efficiency.

C. Ethical Implications of Autonomous Defense: Accountability and Auditability

The deployment of an autonomous, cryptographically enforced security framework raises important ethical questions regarding accountability and auditability. In the proposed system, the VRF jury selection is deterministic and publicly verifiable, meaning that any node can audit the selection process after the fact. This transparency is a significant advantage over traditional centralized security mechanisms, where the selection of monitoring nodes is opaque and potentially biased. However, the predator-prey anomaly detector operates as a black box: the population dynamics are governed by parameters that are set during initialization and are not easily interpretable by human operators. If the detector produces a false positive that causes the routing controller to isolate a legitimate node, the affected node's operator may have difficulty understanding why the decision was made.

To address this accountability gap, the framework could be augmented with an explainability module that generates human-readable justifications for each anomaly verdict. For example, the module could output a natural language explanation such as "Node 42 was flagged because the predator population for flow F3 exceeded the threshold of 0.7, driven by a 40% increase in packet inter-arrival time variance over the last 50 time steps." This explanation could be generated by a rule-based system that maps the internal state of the predator-prey model to a set of predefined templates. While such explanations would not capture the full complexity of the ecological dynamics, they would provide sufficient transparency for human operators to verify the system's behavior.

Another ethical concern is the potential for the framework to be used as a tool for censorship or traffic discrimination. The routing controller's reward function penalizes nodes with high anomaly scores, which could be exploited by a malicious operator to systematically exclude traffic from certain sources. For example, an operator could artificially inflate the anomaly scores of nodes belonging to a competitor's network, causing the routing controller to avoid those nodes and degrade the competitor's service quality. To prevent this, the VRF jury selection and the anomaly detection process must be fully transparent and auditable by independent third parties. The cryptographic proofs generated by the VRF engine provide a foundation for this auditability, but the predator-prey parameters and the OC-SVM model must also be publicly disclosed and verifiable. Future work could explore the use of zero-knowledge proofs to allow the jury to prove that its anomaly scores were computed correctly without revealing the underlying population states, thereby preserving both transparency and privacy.

D. Future Research Directions

Building on the discussion above, we identify several promising directions for future research. First, the integration of a verifiable delay function into the seed derivation process would enhance the framework's resilience against seed manipulation attacks, as discussed in Section 6.2. Second, the development of a lightweight, distributed version of the DQN routing controller would enable fully decentralized operation without a central controller, using techniques such as federated reinforcement learning [28] or multi-agent deep reinforcement learning [29]. Third, the predator-prey anomaly detector could be extended to incorporate multiple predator species, each specialized for a different type of attack (e.g., one predator for packet injection, another for sinkhole attacks, and a third for data manipulation). This multi-species approach would improve detection specificity and reduce false positive rates.

Fourth, the framework's performance under adversarial machine learning attacks should be evaluated. An adversary could attempt to craft traffic patterns that mimic normal behavior while still achieving malicious goals, thereby evading the OC-SVM baseline detector. The predator-prey dynamics provide some resilience against such evasion, as the temporal accumulation of subtle anomalies can still shift the ecological equilibrium. However, a determined adversary with knowledge of the predator-prey parameters could potentially design an attack that maintains the populations within the normal range. Future work could explore the use of adversarial training to harden the OC-SVM and the predator-prey model against such attacks.

Fifth, the framework could be extended to support multi-hop routing decisions, where the DQN controller selects a complete path from source to destination rather than just the next hop. This would require a larger action space and a more complex state representation, but it could lead to more globally optimal routing decisions. The use of graph neural networks [30] to encode the network topology and threat map could enable the DQN to generalize across different network configurations.

Finally, a real-world deployment of the framework in a testbed environment, such as a smart grid or an industrial IoT network, would provide valuable insights into its practical performance and operational challenges. Such a deployment would require careful consideration of hardware constraints, network latency, and integration with existing protocol stacks. The results of this deployment would inform the refinement of the framework's parameters and the development of deployment guidelines for practitioners.

VII. Conclusion

This paper has presented VRF-Jury and Swarm-Secured Routing, a nature-inspired cybersecurity framework for decision support systems that integrates cryptographic randomness, ecological dynamics, and reinforcement learning into a closed-loop security architecture. The framework addresses the fundamental limitation of static trust anchors in distributed routing environments by introducing a verifiable,

non-deterministic jury selection mechanism based on Verifiable Random Functions. This selection is unpredictable, unbiased, and publicly auditable, ensuring that no fixed node becomes a persistent attack surface. The cryptographically elected jury then executes a novel anomaly detection algorithm modeled on Lotka-Volterra predator-prey dynamics, which provides adaptive, decentralized threat identification without requiring centralized training or global state. The temporal memory inherent in the ecological model enables the detection of slow-drift attacks that evade static threshold-based approaches. The final component is a deep Q-network routing controller that jointly optimizes for transmission performance and security, dynamically rerouting traffic around nodes flagged by the jury's collective verdict. The experimental evaluation, conducted across three attack scenarios—packet injection, sinkhole attacks, and gradual denial-of-service escalation—demonstrated that the proposed framework achieves a detection rate of up to 94.7% with a false positive rate of only 3.2%, while maintaining a packet delivery ratio of 92.5% and the lowest average end-to-end latency among all compared methods. The ablation study confirmed that each component contributes meaningfully to the overall performance, with the predator-prey dynamics providing the largest improvement in detection sensitivity. The scalability analysis showed that the logarithmic jury size keeps communication overhead sub-linear, and the framework maintains stable detection rates across networks ranging from 50 to 500 nodes. The primary practical implication is that cryptographically verifiable oversight, combined with bio-inspired adaptation and learning-based routing, can provide a self-healing security layer for decision support systems without centralized coordination or fixed trust assumptions. The framework eliminates single points of failure, resists adaptive adversaries through continuous jury rotation, and enables transparent auditability of all security-critical decisions. These properties make it particularly suitable for dynamic, resource-constrained environments such as sensor networks, smart grid infrastructure, and industrial IoT systems, where static security mechanisms are insufficient against evolving threats. While challenges remain in computational overhead for severely constrained devices and in resilience against seed manipulation attacks, the framework establishes a foundation for future research into integrated, verifiable, and adaptive cybersecurity architectures for distributed decision support systems.

VIII. References

[1] R. Bonczek, C. Holsapple, and A. Whinston, "Foundations of decision support systems," books.google.com, 2014.

[2] D. Power, "Decision support systems: A historical overview," *International Handbooks on Information Systems*, 2008.

[3] A. Perrig, R. Szewczyk, V. Wen, D. Culler, *et al.*, "SPINS: Security protocols for sensor networks," in *Proceedings of the 7th annual international conference on mobile computing and networking*, 2001.

[4] S. Goldberg, J. Vcelak, D. Papadopoulos, and L. Reyzin, "Verifiable random functions (VRFs)," open.bu.edu, 2018.

[5] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, *et al.*, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th ACM symposium on operating systems principles*, 2017.

[6] A. Ewees, F. Ismail, R. Ghoniem, and M. Gaheen, "Enhanced marine predators algorithm for solving global optimization and feature selection problems," *Mathematics*, 2022.

[7] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, 2002.

[8] S. Thrun and M. Littman, "Reinforcement learning: An introduction." *AI Magazine*, 2000.

[9] C. Do, N. Tran, C. Hong, C. Kamhoua, *et al.*, "Game theory for cyber security and privacy," *ACM Computing Surveys*, 2017.

[10] P. D'haeseleer, S. Forrest, *et al.*, "An immunological approach to change detection: Algorithms, analysis and implications," in *Proceedings*, 1996.

[11] M. Islam and M. Merlec, "Proof of random leader: A fast and manipulation-resistant proof-of-authority consensus algorithm for permissioned blockchains using verifiable random function," *IEEE Transactions on Services Computing*, 2025.

[12] H. Wang, X. Liu, and J. Chen, "RVR blockchain consensus: A verifiable, weighted-random, byzantine-tolerant framework for smart grid energy trading," *Computers*, 2025.

[13] Y. Zhuang, Y. Chen, X. Zhang, T. Ren, *et al.*, "A large-scale node lightweight consensus algorithm of blockchain for internet of things," *IEEE Internet of Things Journal*, 2024.

[14] C. Rammohan, P. Laxmikanth, D. Srikar, *et al.*, "The BioShield algorithm: Pioneering real-time adaptive security in IoT networks through nature-inspired machine learning," *International Journal of Electrical and Computer Engineering*, 2023.

[15] S. Qureshi and S. Shandilya, "Nature-inspired adaptive decision support system for secured clustering in cyber networks," *Multimedia Tools and Applications*, 2024.

[16] S. Qureshi, S. Padhy, P. Suman, and A. Suman, "Hybrid secure clustered nature-inspired decision support system for secure data communication in cyberspace," *Cyberology*, 2025.

[17] N. Wagner, C. Sahin, J. Pena, *et al.*, "A nature-inspired decision system for secure cyber network architecture," in *2017 IEEE symposium series on computational intelligence (SSCI)*, 2017.

[18] E. Bouzidi, A. Outtagarts, R. Langar, *et al.*, "Deep q-network and traffic prediction based routing optimization in software defined networks," *Journal of Network and Computer Applications*, 2021.

[19] V. Gupta, S. Shandilya, C. Ganguli, *et al.*, "Data-driven approach to network intrusion detection system using modified artificial bee colony algorithm for nature-inspired

cybersecurity," in *International conference on machine learning and data engineering*, 2023.
[20] G. Lahbar, Z. Guanghai, D. Prihadi, *et al.*, "SQDeChain: Enhancing tourist trust and destination image through a hybrid AI-blockchain framework for tourism 5.0," *Information Technology & Tourism*, 2026.
[21] V. Volterra, "13 fluctuations in the abundance of a species considered mathematically," *Foundations of Ecology*, 2012.
[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, *et al.*, "Human-level control through deep reinforcement learning," *nature*, 2015.
[23] B. Schölkopf, J. Platt, J. Shawe-taylor, A. Smola, *et al.*, "Estimating the support of a high-dimensional distribution, 2001, vol. 13, no. 7," *Neural Computation, issn*, 2026.
[24] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," rfc-editor.org, 2003.
[25] H. Hu, Y. Han, H. Wang, M. Yao, and C. Wang, "Trust-aware secure routing protocol for wireless sensor networks," *ETRI Journal*, 2021.
[26] A. Biryukov, B. Fisch, G. Herold, D. Khovratovich, *et al.*, "Cryptanalysis of algebraic verifiable delay functions," *Lecture Notes in Computer Science*, 2024.
[27] M. Castro and B. Liskov, "Practical byzantine fault tolerance," *OsDI*, 1999.
[28] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated reinforcement learning: Techniques, applications, and open challenges," arXiv preprint arXiv:2108.11887, 2021.
[29] W. LoweR and A. TAMAR, "Multi□ agentactor□ criticfor mixedcooperative□ competitiveenvironments," … (NIPS). LongBeach, USA, 2017, 2017.
[30] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications. cite," arXiv preprint arxiv:1812.08434, 2018, 1812.