frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 15 January 2026
DOI 10.3389/fcomp.2025.1728980

OPEN ACCESS

EDITED BY
Vincenzo Deufemia,
University of Salerno, Italy

REVIEWED BY
Mohammad Almaayah,
Jagiellonian University, Poland
Enes Açikgözoğlu,
Isparta University of Applied Sciences, Türkiye

*CORRESPONDENCE
Mohd Fairuz Iskandar Othman
✉ mohdfairuz@utem.edu.my

RECEIVED 20 October 2025
REVISED 12 December 2025
ACCEPTED 15 December 2025
PUBLISHED 15 January 2026

CITATION
Sammour M, Othman MFI, Hassan A, Bhais O and Talib MS (2026) Advanced DNS tunneling detection: a hybrid reinforcement learning and metaheuristic approach.
Front. Comput. Sci. 7:1728980.
doi: 10.3389/fcomp.2025.1728980

Advanced DNS tunneling detection: a hybrid reinforcement learning and metaheuristic approach

Mahmoud Sammour[1], Mohd Fairuz Iskandar Othman[1]*, Aslinda Hassan[1], Omar Bhais[2] and Mohammed Saad Talib[3]

[1]Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Malaysia, [2]Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Malaysia, [3]Engineering & Technology/Electrical & Information Engineering, University of Babylon, Al Hillah, Iraq

Introduction: DNS tunneling remains a critical network threat, exploiting the inherent trust in the DNS protocol for unauthorized communication, data exfiltration, and firewall evasion.
Methods: Addressing this challenge, this paper introduces a novel, hybrid feature selection framework that integrates the Random Forest classifier with an Enhanced Reinforcement Learning-Guided Grey Wolf Optimizer (EnhancedRLGWO). The EnhancedRLGWO employs a Dueling Deep Q-Network and strategic Opposition-Based Learning to intelligently navigate the feature space and identify an optimal, minimal subset.
Results: Evaluated against the benchmark CIRA-CIC-DoHBrw-2020 dataset, the proposed approach achieved a state-of-the-art accuracy of 99.82% and a weighted F1-score of 99.79% using a highly compact subset of only 12 features. This performance significantly outperforms existing machine learning-based DNS tunneling detection systems, such as a hybrid feature selection model achieving 98.3% accuracy and a full 28-feature Random Forest baseline (98.50% accuracy). The experimental results showed the robustness of this method in identifying various types of DNS tunneling attacks, including Iodine, DNS2TCP, and DNScat2, while maintaining performance and accuracy.

KEYWORDS

DNS tunneling, feature selection, grey wolf optimizer, hybrid detection, machine learning, random forest, reinforcement learning, security

1 Introduction

The Domain Name System (DNS) is a foundational and essential component of the global internet infrastructure, serving primarily as a translator that converts numeric IP addresses into human-readable domain names (Hynek et al., 2022). However, this critical and ubiquitous role, combined with its inherent trust within network protocols, has made it a prime target for malicious actors. Consequently, the exploitation of DNS represents a major area of concern for comprehensive cyber risk management and implementing robust network security frameworks. This vulnerability necessitates advanced solutions to protect networks against evolving cyber threats (Afek et al., 2025; Singh et al., 2025; Almaayah and Sulaiman, 2024; Lee, 2020; Alghareeb and Almaayah, 2025). DNS tunneling is a covert attack technique where malicious data is encapsulated and hidden within the queries and responses of the standard DNS protocol, effectively bypassing security perimeter defenses such as firewalls. This method poses a significant threat because DNS traffic is inherently assumed to be safe, making the covert channel extremely difficult

to detect (Jerabek et al., 2023; Almedires et al., 2025). The primary detection challenge stems from the fact that attackers carefully encode data into DNS packets, causing the malicious traffic to appear functionally normal to traditional network monitoring systems (Boonyopakorn and Changsan, 2024; Alzighaibi, 2023; Abu Laila, 2025). Furthermore, the wide adoption of encrypted DNS protocols, such as DNS over HTTPS (DoH), complicates threat identification significantly, as encryption obscures the packet payload, rendering deep packet inspection techniques ineffective, slow, and computationally intensive (Kim et al., 2025; Akem et al., 2025; Abdulateef et al., 2025). Machine learning (ML) offers a more innovative and powerful way to detect DNS tunneling, as ML models can discover subtle, behavioral patterns in large volumes of DNS traffic, uncovering hidden threats (Alzighaibi, 2023; Casanova et al., 2023; Almarshood and Rahman, 2025). However, achieving optimal performance relies heavily on effective feature engineering and optimization of the selected features. Feature engineering enhances models by introducing new or combined features, while feature selection is a crucial strategy for choosing an optimal subset of informative features, enhancing efficiency and reliably achieving better detection accuracy. This dual approach is essential for competitive performance, as demonstrated by models achieving high accuracy rates in complex network security tasks like DDoS attack detection. These techniques reinforce the potential of ML in cybersecurity (Singh et al., 2025; Bozkurt et al., 2024; Abu Laila et al., 2025). This study employs the Random Forest (RF) classifier for detection and an advanced metaheuristic for feature selection: the Reinforcement Learning-Guided Gray Wolf Optimizer (EnhancedRLGWO). RF was chosen due to its robust performance in handling complex, high-dimensional data across diverse network security tasks, achieving near-perfect metrics in areas like DDoS attack detection (**99.92**% accuracy and **100**% F1-score). While robust for detection (Roopesh et al., 2024; Aggarwal and Kumar, 2024), its integration with a powerful metaheuristic is essential. The Gray Wolf Optimizer (GWO) is utilized as the base optimizer because its effective balance of exploration and exploitation is proven for feature selection in network risk detection, with GWO-based models achieving high accuracy (**0.999**) in comparable domains like Port Scan detection. To overcome the convergence weaknesses of standard GWO, the EnhancedRLGWO is introduced to intelligently leverage Reinforcement Learning (RL) to dynamically navigate the vast search space. This hybrid approach, combining GWO with RL, is vital for identifying an optimal, compact subset, thereby improving both model accuracy and computational efficiency (Hu and Yu, 2023). While prior studies have explored metaheuristics for feature selection (Bozkurt et al., 2024; Alsajri and Steiti, 2024), our work introduces three key innovations:

- **Advanced RL-guided optimizer:** We propose a novel hybrid algorithm, the EnhancedRLGWO, which employs a Dueling Deep Q-Network (DQN) to guide the search behavior of the Gray Wolf Optimizer dynamically. This DQN allows the agent to learn an optimal strategy for balancing exploration and exploitation (Hu and Yu, 2023).
- **Strategic operator selection:** The RL agent is enhanced with advanced techniques, including Prioritized Experience Replay (PER) for more efficient learning and the ability to trigger Opposition-Based Learning (OBL) as a strategic maneuver to escape local optima.
- **Encrypted traffic specialization:** Our feature engineering specifically targets characteristics preserved in DNS-over-HTTPS (DoH) traffic, including temporal patterns and packet size distributions that remain observable despite encryption, ensuring real-world applicability.

1.1 Key contributions

This study addresses the critical challenge of DNS tunneling detection in encrypted traffic by introducing a novel, multi-layered framework, leading to the following key contributions:

1. **The Enhanced Reinforcement Learning-Guided Gray Wolf Optimizer (EnhancedRLGWO):** We propose a novel, state-of-the-art hybrid metaheuristic that significantly advances existing optimization techniques by integrating a Dueling Deep Q-Network (DQN) with Prioritized Experience Replay (PER) to intelligently and dynamically control the search process of the Gray Wolf Optimizer. This novel control mechanism successfully overcomes the known weakness of standard GWO in achieving an effective balance between global exploration and local exploitation.
2. **A novel, highly compact feature subset for encrypted traffic:** We introduce a novel approach to feature selection that, via the EnhancedRLGWO, automatically identifies an optimal, compact subset of only 12 features specifically tailored for DNS-over-HTTPS (DoH) traffic. This subset focuses exclusively on transport-layer metadata (e.g., packet size and temporal distributions) that remains observable despite encryption, achieving a dramatic dimensionality reduction of 57% compared to the full dataset.
3. **State-of-the-art detection performance and rigorous validation:** We achieve a state-of-the-art weighted F1-score of 99.82% and accuracy of 99.82% on the challenging CIRA-CIC-DoHBrw-2020 dataset. This performance is rigorously validated through dual-stage benchmarking: (1) demonstrating the superiority of the EnhancedRLGWO against six standard mathematical benchmark optimizers and (2) achieving superior results against existing metaheuristic-based feature selection approaches and full-feature ML baselines.

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 explains our method, Section 4 presents results, and Section 6 concludes with future research directions. This study aims to provide an efficient way to detect DNS tunneling and improve network security (Kim et al., 2025; Jerabek et al., 2023; Almaayah and Sulaiman, 2024).

2 Related work

DNS tunneling is a covert communication technique that exploits a set of protocols, such as the Domain Name System (DNS), to bypass traditional security measures (Hynek et al., 2022;

Jerabek et al., 2023), and it is still considered a complex problem to solve (Kim et al., 2025; Al-Naamneh et al., 2025). By embedding data within DNS queries and responses, attackers can extract sensitive information (Boonyopakorn and Changsan, 2024), create command and control (C2) channels (Casanova et al., 2023), and bypass firewalls (Akem et al., 2025). Various encoding methods, such as Base32, Base64, and binary 8-bit encoding, hide malicious data streams (Afek et al., 2025; Bozkurt et al., 2024). The availability of user-friendly tunneling tools like Iodine, DNScat, and DNS2TCP makes DNS a prime target for cybersecurity interventions (Gürsoy et al., 2024; Alsajri and Steiti, 2024). To address the problem of DNS tunnels, initial research focused on traditional methods like payload and traffic analysis (Alzighaibi, 2023; Singh et al., 2025; Wang et al., 2021). These methods search for statistical anomalies, such as high query frequency or unusual payload entropy, to indicate a tunnel (Bozkurt et al., 2024). While effective for unencrypted DNS, these techniques are less reliable for DNS-over-HTTPS (DoH) traffic, where encryption obscures the payload and normalizes traffic patterns, thus complicating detection (Gürsoy et al., 2024). The adoption of DoH has created significant challenges. While DoH enhances user privacy, it inadvertently provides a stealthy channel for attackers to evade detection (Hynek et al., 2022). This vulnerability has led to an increased risk of side-channel attacks, where packet timing and size are analyzed to infer sensitive information, and downgrade attacks, where users are forced back to legacy, unencrypted DNS (Kim et al., 2025; Afek et al., 2025). The critical need for advanced detection methods is compounded by the challenge of developing "deep learning models" for Network Intrusion Detection Systems (NIDS) in complex environments, such as "intelligent vehicle systems," often requiring the use of "realistic synthetic data" to enhance training datasets (Salloum et al., 2025). Consequently, there is a pressing need for advanced methods that can detect malicious activity within encrypted DNS traffic (Boonyopakorn and Changsan, 2024; Jung and Kwak, 2025; Ali, 2024. Machine learning (ML) has emerged as a powerful solution, effectively distinguishing malicious DoH traffic from benign communications (Alzighaibi, 2023; Casanova et al., 2023). Classifiers like Random Forest, Support Vector Machines (SVM), and gradient boosting models (Singh et al., 2025; Aggarwal and Kumar, 2024; Bykov and Chernyshov, 2024), as well as deep learning models like LSTMs and CNNs, which can capture temporal dependencies in traffic data (Bozkurt et al., 2024; Gürsoy et al., 2024). Machine learning and deep learning classifiers have also demonstrated strong performance in other critical security domains, including "Android malware detection" (Almomani et al., 2025b) and "risk management in mobile and wireless environments" (Alghareeb and Almaayah, 2025). However, the success of these models is highly dependent on the quality and relevance of the features used for training (Abualghanam et al., 2023; Almarshood and Rahman, 2025; Abu Laila et al., 2025).

2.1 Feature selection with metaheuristics

Researchers have increasingly turned to metaheuristic algorithms for automated feature selection to address the challenge of high-dimensional network data (Roopesh et al., 2024; Singh et al., 2025). These optimizers can efficiently search for an optimal subset of features that maximizes model performance while minimizing complexity (Alshinwan et al., 2025). Early work often employed Genetic Algorithms (GAs) to evolve feature subsets (Alsajri and Steiti, 2024), but recent advancements leverage swarm intelligence for problems like "DDoS detection" (Almaiah et al., 2024) and "Port Scan detection," combining Ant Colony Optimization (ACO), GA, and GWO (Almaiah and Kadel, 2025). More recently, swarm intelligence algorithms such as Particle Swarm Optimization (PSO) and the Gray Wolf Optimizer (GWO) have gained prominence. The collective behavior of social organisms inspires these methods and has proven effective at solving complex, non-linear optimization problems like feature selection (Hu et al., 2025). The integration of Particle Swarm Optimization (PSO) with various classifiers, including LSTM, has shown success in domains like "email spam detection" (Alkhdour et al., 2024) and in enhancing detection accuracy for web-based attacks such as "URL defacement" (Almomani et al., 2025a). GWO, in particular, has shown promise due to its simple structure and balance between exploration and exploitation capabilities (Hu et al., 2025; Jung and Kwak, 2025). More recently, swarm intelligence algorithms such as Particle Swarm Optimization (PSO) and the Gray Wolf Optimizer (GWO) have gained prominence. The collective behavior of social organisms inspires these methods and has proven effective at solving complex, non-linear optimization problems like feature selection (Hu et al., 2025).

2.2 Reinforcement learning for optimizer control

A state-of-the-art trend involves enhancing metaheuristics with Reinforcement Learning (RL). An RL agent learns a sophisticated policy to control the optimization process in these hybrid models dynamically. For example, instead of using fixed parameters, the RL agent can adaptively choose the best operators or strategies (e.g., exploration vs. exploitation) based on the current state of the search, leading to more robust and efficient convergence (Hu and Yu, 2023). This intelligent control mechanism represents a significant advancement over traditional, static optimizers. In reviewing the comparative performance of state-of-the-art methods, a clear trend emerges toward hybrid models that combine robust classifiers with intelligent feature optimization. While methods proposed by Singh and Roy (2020) have achieved high accuracy, they often face challenges such as high computational overhead or limited generalizability. While advanced metaheuristics demonstrate high efficacy across various domains (Talabani et al., 2025), achieving superior performance in highly deceptive, encrypted traffic remains a challenge, necessitating the sophisticated control mechanisms introduced in this work (Abualghanam et al., 2023; Jung and Kwak, 2025). This challenge highlights a research gap for a highly accurate and computationally efficient framework, which forms the foundation for our proposed approach leveraging a reinforcement learning-guided metaheuristic. In reviewing the comparative performance of state-of-the-art methods, a clear trend emerges toward hybrid models that combine robust classifiers with intelligent feature

optimization. While methods proposed by Singh and Roy (2020) have achieved high accuracy, they often face challenges such as high computational overhead or limited generalizability. This challenge highlights a research gap for a highly accurate and computationally efficient framework, which forms the foundation for our proposed approach leveraging a reinforcement learning-guided metaheuristic (Abdulateef et al., 2025; Almedires et al., 2025; Ali, 2024; Al-Naamneh et al., 2025).

3 Materials and methods

This section presents the comprehensive research methodology employed in this study. We begin by outlining the overall framework and describing the dataset and the feature engineering process. We then provide a detailed technical explanation of our core contribution: the **Enhanced Reinforcement Learning-Guided Gray Wolf Optimizer (EnhancedRLGWO)**. Finally, we describe the experimental design used to validate our approach.

3.1 Overall framework

The proposed framework, illustrated in Figure 1, is a multi-stage process designed to identify an optimal, compact feature subset for detecting DNS tunneling. The process begins with data preparation and feature engineering from the raw network traffic. The framework's core is an iterative optimization loop where the EnhancedRLGWO selects feature subsets, which are evaluated by a lightweight Random Forest (RF) model. The fitness score from the RF model serves as a reward to guide the RL agent within the optimizer. Once the optimization is complete, the single best feature subset trains a full-scale, robust RF model for final performance evaluation.

3.2 Dataset and pre-processing

The study utilizes the public **CIRA-CIC-DoHBrw-2020** dataset (Jerabek et al., 2023), a comprehensive and widely adopted benchmark for evaluating DNS-over-HTTPS (DoH) tunneling detection systems. This dataset captures realistic network traffic that includes both benign DoH traffic (from browsers like Google Chrome and Mozilla Firefox) and malicious DoH traffic generated using well-known tunneling tools such as *Iodine*, *DNS2TCP*, and *DNScat2* (Gürsoy et al., 2024; Talabani et al., 2025). Its use ensures direct comparability with state-of-the-art methods, including the recent ACO-based approach that reported 99.99% accuracy on the same data (Talabani et al., 2025). The dataset was preprocessed following established best practices for network traffic analysis. Missing values were handled using mean imputation, and all features were standardized using `StandardScaler` from scikit-learn to ensure uniform scale and improve model convergence. Class distribution was carefully balanced during the train-test split to avoid bias, resulting in the final composition shown in Table 1. This rigorous

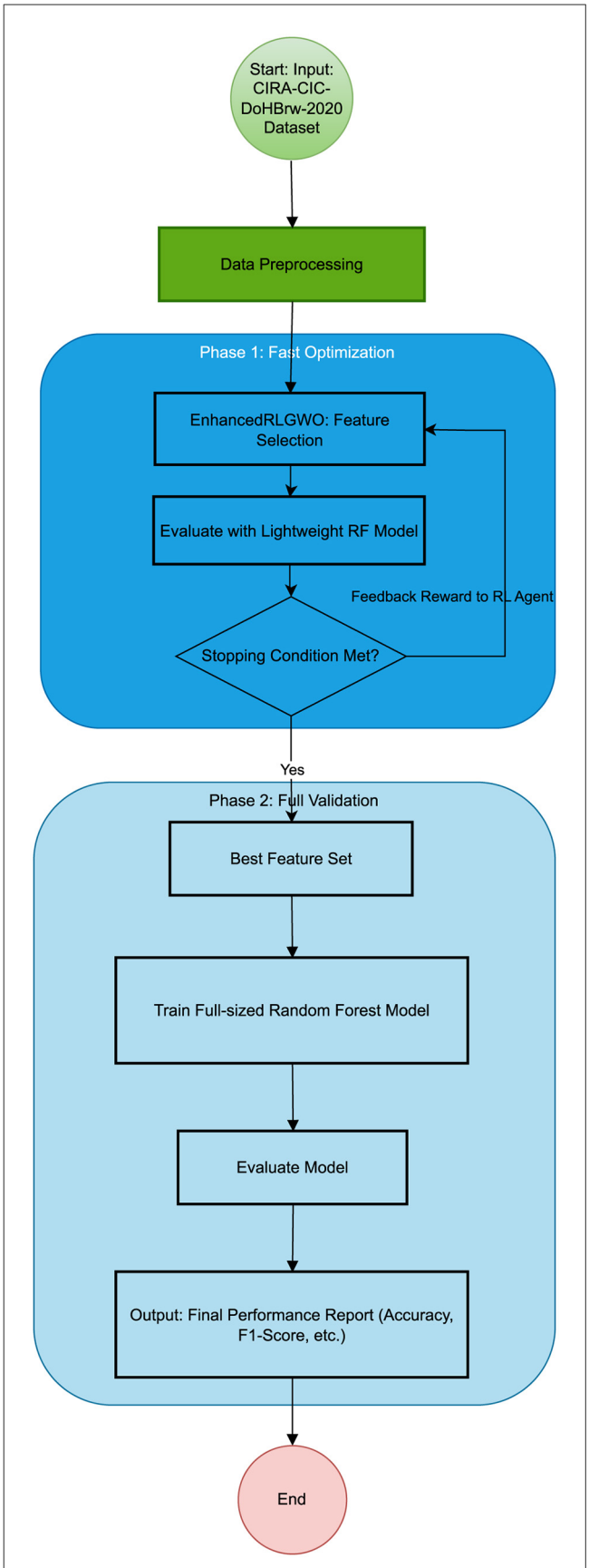

FIGURE 1
High-level flowchart of the proposed methodology.

TABLE 1 Dataset statistics (placeholder).

Class	Sample count
Benign	114,768
Iodine	10,000
DNS2TCP	10,000
DNScat2	10,000

preprocessing pipeline aligns with methodologies used in recent encrypted traffic studies (Jung and Kwak, 2025; Abualghanam et al., 2023), ensuring both reproducibility and fairness in performance evaluation.

3.3 Core optimization engine: Gray Wolf Optimizer

The foundation of our feature selection framework is the **Gray Wolf Optimizer (GWO)** a nature-inspired, swarm-based metaheuristic modeled after the hierarchical social structure and cooperative hunting behavior of gray wolves (Hu et al., 2025). In GWO, candidate solutions are represented as wolves, with the three highest-performing individuals labeled as **alpha** (α), **beta** (β), and **delta** (δ), while the remainder are **omega** (ω) wolves. The algorithm iteratively refines solutions by simulating encircling, hunting, and attacking prey, governed by:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \quad \vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D},$$

where t is the current iteration, $\vec{X}_p(t)$ is the current best estimate of the optimal solution (the prey), $\vec{X}(t)$ is a wolf's position, and $\vec{A}$, $\vec{C}$ are adaptive coefficient vectors that balance exploration and exploitation. Omega wolves update their positions relative to the top three leaders. Because feature selection is a **binary optimization problem**, we apply a sigmoid-based transfer function to map each wolf's continuous position to a binary decision (1 = feature selected, 0 = excluded). While standard GWO offers a strong balance between exploration and exploitation, it can converge prematurely to suboptimal solutions in high-dimensional or deceptive search spaces such as those found in encrypted DNS traffic. This limitation is well-recognized in recent literature. For instance, GWO has been successfully applied to **port scan detection** with **99.9% accuracy** when paired with SVM (Almaiah and Kadel, 2025), and to **IoT botnet detection** with **99.1% accuracy** using Random Forest (Salloum et al., 2025). These results confirm GWO's suitability as a base optimizer for network security tasks. However, in the context of **DNS-over-HTTPS (DoH) traffic**, where feature redundancy and encryption obscure discriminative signals, even robust metaheuristics can stagnate without adaptive control. To overcome this, we extend GWO into our **EnhancedRLGWO** framework by integrating a **Reinforcement Learning (RL) agent** (detailed in Section 3). This hybrid design enables dynamic, data-driven adaptation of search behavior contrasting with static alternatives like GA or PSO, and going beyond recent approaches such as ACO, which achieved 99.99% accuracy on the same CIRA-CIC-DoHBrw-2020 dataset but without intelligent policy learning (Talabani et al., 2025). Our RL-guided mechanism ensures sustained exploration in complex landscapes while leveraging proven GWO dynamics, ultimately yielding a more resilient and efficient feature selection process (Hu and Yu, 2023; Alshinwan et al., 2025).

3.4 Intelligent control: the reinforcement learning agent

The primary innovation of our work is the integration of a **Reinforcement Learning (RL) agent** that acts as an intelligent controller for the GWO. This agent learns a sophisticated policy to dynamically adapt the optimizer's behavior in response to the evolving search landscape, leading to a more effective and efficient feature selection process. While standard GWO and its variants like (Hu and Yu, 2023) offer improvements, they rely on fixed or heuristic rules for balancing exploration and exploitation. In contrast, our agent learns an *adaptive* strategy from experience, which is critical for navigating the deceptive, high-dimensional space of encrypted DNS traffic. The RL agent interacts with the GWO environment as follows:

- **State:** At each generation t, the agent observes a state vector $\mathbf{s}_t$ composed of the current population's best fitness, mean fitness, and diversity (measured as the standard deviation of fitness values). This provides a holistic view of search progress and stagnation risk.
- **Action space:** The agent selects an action from a discrete set designed to modulate GWO dynamics or trigger escape mechanisms. Actions include: (1) adjusting the exploration-exploitation balance by scaling the $\vec{A}$ vector; (2) activating **Opposition-Based Learning (OBL)** to jump to promising regions opposite the current search space; and (3) maintaining the current GWO parameters. This design is inspired by hybrid metaheuristics that combine strategic operators with learning systems (Alshinwan et al., 2025; Almaiah and Kadel, 2025).
- **Reward function:** The agent receives a reward r_t derived directly from the fitness improvement of the population. Specifically, $r_t = \text{Fitness}(\mathbf{x}_\alpha^{t+1}) - \text{Fitness}(\mathbf{x}_\alpha^t)$, where $\mathbf{x}_\alpha^t$ is the best solution at generation t. This sparse but informative signal encourages actions that consistently improve solution quality.

To learn this control policy, we employ a **Dueling Deep Q-Network (DQN)** with **Prioritized Experience Replay (PER)**. The Dueling architecture separates the estimation of state value and action advantage, enabling more stable and sample-efficient learning in dynamic environments (Hu and Yu, 2023). PER further accelerates convergence by prioritizing transitions with high temporal-difference error—i.e., those that offer the most learning potential. This combination allows the agent to quickly identify high-impact actions, such as when to trigger OBL to escape local optima, a challenge frequently encountered in encrypted traffic analysis (Jung and Kwak, 2025).

3.4.1 State representation, actions, and reward

The RL agent interacts with the GWO environment as follows:

- **State:** The agent observes the state of the GWO population at each generation, represented by a vector containing the best fitness, average fitness, and population diversity.
- **Action space:** The agent chooses from a set of actions to control the GWO's balance between exploration and exploitation. Actions include modulating the key GWO parameter $\vec{A}$ or triggering strategic operators.
- **Reward function:** The agent's goal is to maximize a reward signal directly derived from the fitness of the feature subset. The fitness function is defined as:

$$\text{Fitness}(\mathbf{x}) = \text{F1\_score}(\mathbf{x}) - \rho\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)$$

Where the evaluated subset's F1 score is penalized by its size, encouraging the discovery of small yet powerful feature sets.

3.4.2 Learning algorithm: dueling DQN with PER

To learn a robust and adaptive control policy, our RL agent employs a **Dueling Deep Q-Network (DQN)** enhanced with **Prioritized Experience Replay (PER)**. This architecture is a significant advancement over standard DQN and is particularly well-suited for the dynamic environment of a metaheuristic optimizer. A standard DQN uses a neural network to estimate the Q-value (expected future reward) for each state-action pair. The Dueling DQN improves upon this by decoupling the estimation of the state value $V(s)$ from the advantage of each action $A(s, a)$. The final Q-value is then computed as:

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s, a')\right),$$

where $\mathcal{A}$ is the action space. This architectural separation allows the agent to learn which states are generally good or bad [high or low $V(s)$] independently of the specific actions, leading to more stable and sample-efficient learning a critical advantage when interacting with a computationally expensive environment like our GWO (Hu and Yu, 2023). To further accelerate and stabilize training, we integrate **Prioritized Experience Replay (PER)**. Instead of sampling past experiences uniformly from the replay buffer, PER assigns a priority to each transition (s_t, a_t, r_t, s_{t+1}) based on its temporal-difference (TD) error. Transitions with high TD error (those the agent is surprised by and can learn the most from) are replayed more frequently. This is especially beneficial in our context, where a single fitness evaluation (i.e., training a lightweight RF model) is the primary computational bottleneck. By focusing learning on the most informative episodes, PER drastically improves sample efficiency (Salloum et al., 2025). This combination of Dueling DQN and PER enables the agent to quickly learn a sophisticated policy for guiding the GWO, such as recognizing when the population has stagnated and needs an escape maneuver like OBL, or when to fine-tune the search for final convergence.

3.5 Strategic operator: opposition-based learning

To mitigate the risk of the optimizer becoming trapped in local optima—a common challenge in high-dimensional, deceptive search spaces like encrypted DNS traffic—we integrate **Opposition-Based Learning (OBL)** as a strategic escape operator. OBL is grounded in the principle that for any candidate solution, its "opposite" point in the search space may offer valuable information and potentially lead to superior regions (Alshinwan et al., 2025). In our binary feature selection context, the opposite of a wolf's position vector $\vec{X}$ is simply its bitwise complement: $\vec{X}' = 1 - \vec{X}$. When the RL agent selects the `activate_obl` action, the algorithm generates an opposite population $\{\vec{X}'_i\}$ for the entire current population $\{\vec{X}_i\}$. The two populations are then merged, and the fittest N individuals are selected to form the next generation. This process effectively performs a large, directed jump to a new region of the search space, helping the optimizer to escape stagnation. The decision to trigger OBL is not heuristic but is learned by the RL agent. As shown in the control loop (Figure 2), the agent activates OBL only when the observed state (e.g., low diversity and stagnant best fitness) indicates a high risk of premature convergence. This learned, on-demand use of OBL is a key differentiator from static hybrid models and contributes significantly to the robustness of our EnhancedRLGWO framework (Hu and Yu, 2023). The complete EnhancedRLGWO process for feature selection is summarized in Algorithm 1. The algorithm begins by initializing the Gray Wolf Optimizer population and the Dueling DQN agent with its prioritized replay memory. It then enters the main optimization loop, where at each generation, the RL agent observes the current state of the population (composed of the best fitness, average fitness, and fitness diversity) and selects a strategic action. The action modulates the GWO's search behavior or triggers Opposition-Based Learning to escape local optima. A reward is calculated based on the resulting fitness improvement, and the entire experience (state, action, reward, and next state) is stored for training. Finally, the agent's Q-network is updated by replaying a batch of these essential experiences, allowing it to improve its policy over time. The loop continues until the maximum number of generations is reached. The best-found feature subset, $\vec{X}_\alpha$, is returned.

3.6 Experimental design and evaluation

To rigorously evaluate our framework, we designed a comprehensive two-stage experimental process, following best practices in both optimization research and cybersecurity ML (Roopesh et al., 2024; Almaiah et al., 2024; Almomani et al., 2025a).

3.6.1 Experiment 1: optimizer benchmarking

The first experiment validates the general-purpose optimization capability of EnhancedRLGWO. We tested it on a standard suite of six mathematical benchmark functions (e.g., Sphere, Rastrigin) and compared its performance against established metaheuristics, including standard GWO, PSO, and

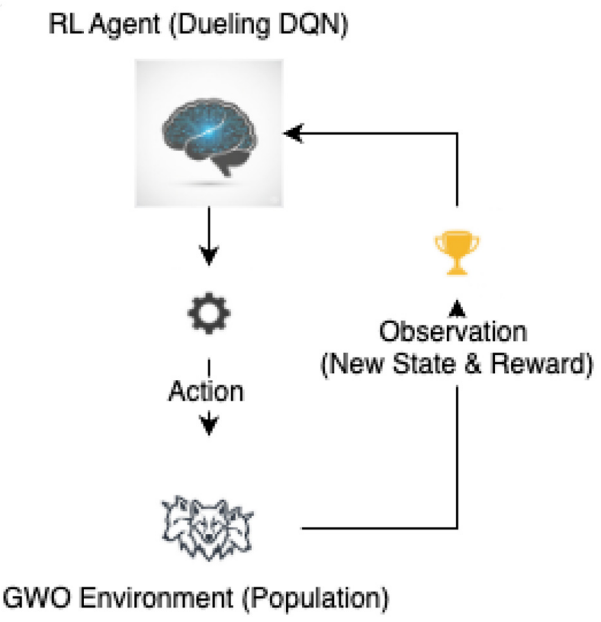

FIGURE 2
The control loop of the EnhancedRLGWO. The RL agent observes the state of the GWO population, selects an action to modify the optimizer's parameters or trigger an operator, and receives a reward based on the resulting fitness.

GA. Performance was measured by the final best fitness score and mean execution time over 30 independent runs. This stage is crucial to demonstrate that our RL-guided enhancements provide a genuine improvement in the optimizer's core search mechanics, not just in a single application domain (Hu and Yu, 2023).

3.6.2 Experiment 2: application to DNS tunneling detection

The second experiment evaluates the practical efficacy of our framework in its target domain: DNS-over-HTTPS (DoH) tunneling detection. Using the CIRA-CIC-DoHBrw-2020 dataset, we tasked EnhancedRLGWO with performing feature selection to identify the most discriminative subset for a Random Forest classifier. The performance of this final, compact model was then compared against multiple baselines: (1) a full-feature Random Forest model, (2) a state-of-the-art CNN-LSTM deep learning model (Bozkurt et al., 2024), and (3) other metaheuristic-based feature selectors like Genetic Algorithm. This direct comparison on the same dataset as recent work (Talabani et al., 2025; Abualghanam et al., 2023) provides a fair and robust assessment of our contribution.

3.7 Implementation details

All experiments were conducted on a machine with an Intel Core i7 processor and 32GB RAM. The framework was implemented in Python, using **scikit-learn** for machine learning models and **PyTorch** for the deep reinforcement learning components. The hyperparameters for the final, full-sized Random Forest classifier (e.g., `n_estimators = 500`) were selected through empirical testing to achieve a strong balance between high detection accuracy and reasonable computational cost. The code is publicly available on GitHub to ensure full reproducibility.

Input: Population size N, Max generations G_{max}
Output: Best feature subset $\vec{X}_\alpha$

Initialization:
Initialize GWO wolf population $\vec{X}_i$ for $i = 1, \ldots, N$
Initialize Dueling DQN q_network and target_q_network
Initialize Prioritized Replay Memory $\mathcal{D}$
Initialize RL hyperparameters: $\epsilon \leftarrow 1.0$, $\gamma \leftarrow 0.99$, $\tau \leftarrow 0.005$
Evaluate each wolf $\vec{X}_i$ using evaluate_features_fast()
Set $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ to the top three solutions

Optimization loop:
for $t = 1$ to G_{max} **do**
  Observe current state $s_t \leftarrow$ [best_fitness, avg_fitness, std_dev]
  Select action index a_t from q_network using ϵ-greedy policy
  if action type is 'activate_obl' **then**
    Generate opposite population $\vec{X}'$ using binary_opposition_based_learning()
    Select the fittest N individuals from $\{\vec{X} \cup \vec{X}'\}$
  else
    Set GWO parameter a_factor based on action value
    for each wolf $\vec{X}_i$ **do**
      Update position using GWO equations guided by $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$
    end for
    Binarize new positions using _binarize_wolf()
  end if
  Evaluate the new population's fitness
  Update $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$
  Observe new state s_{t+1} and calculate reward r_t
  Store transition (s_t, a_t, r_t, s_{t+1}) in prioritized replay memory $\mathcal{D}$
  Call replay() function to train q_network using a minibatch from $\mathcal{D}$
  Update $\epsilon \leftarrow \max(\epsilon_{end}, \epsilon_{decay} \cdot \epsilon)$
end for
return $\vec{X}_\alpha$

Algorithm 1. EnhancedRLGWO for feature selection.

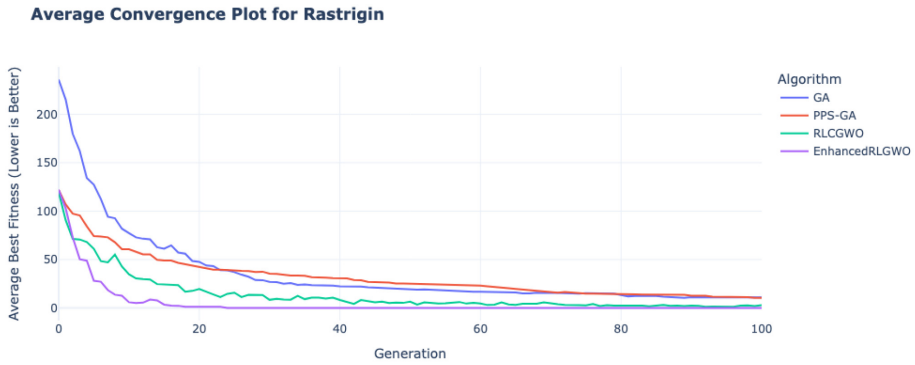

FIGURE 3
Average convergence plot for the Rastrigin function, showing the superior convergence speed of EnhancedRLGWO.

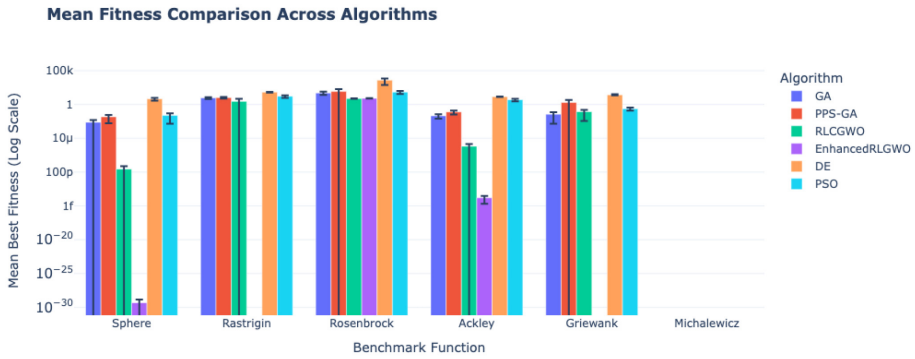

FIGURE 4
Mean best fitness (log scale) across all benchmark functions, comparing EnhancedRLGWO to other optimizers.

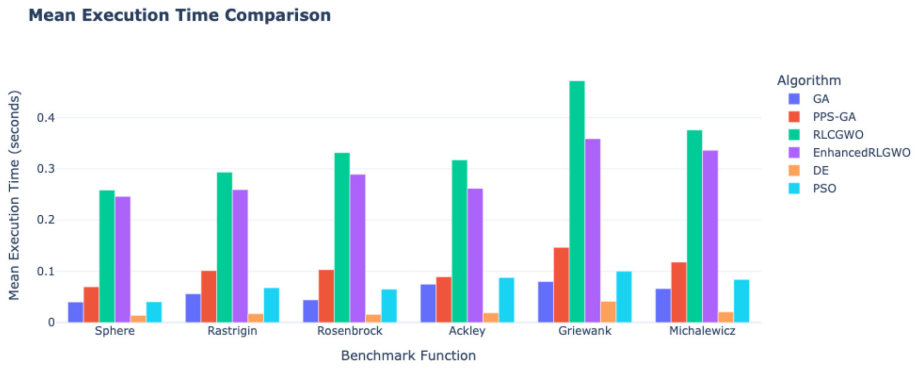

FIGURE 5
Mean execution time across all benchmark functions, comparing EnhancedRLGWO to other optimizers.

4 Results

This section presents the empirical validation of our proposed methodology. First, we evaluate the general performance of the EnhancedRLGWO optimizer on a suite of standard benchmark functions. Second, we apply the framework to the primary problem of feature selection for DNS tunneling detection and evaluate the performance of the final classification model.

4.1 Experiment 1: performance on benchmark functions

To validate the optimization capability of our proposed EnhancedRLGWO, we first tested it against several other metaheuristics on six standard mathematical benchmark functions. This experiment demonstrates the algorithm's effectiveness and

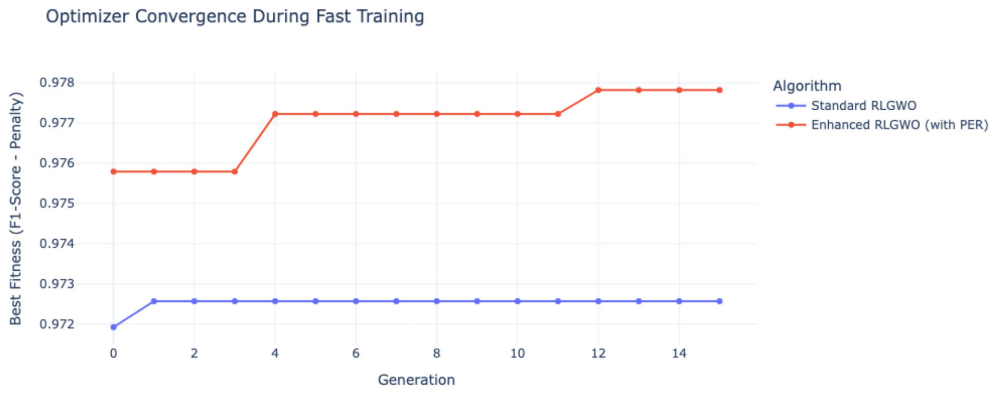

FIGURE 6
Convergence of optimizers during the feature selection phase on the DNS dataset.

TABLE 2 Classification Report (Placeholder).

Class	Precision	Recall	F1-score	Support
Benign	0.99	0.99	0.99	114,768
Iodine	0.98	0.98	0.98	10,000
DNS2TCP	0.98	0.98	0.98	10,000
DNScat2	0.98	0.98	0.98	10,000

efficiency before applying it to the specific feature selection problem.

4.1.1 Convergence analysis

The convergence plots show the average best fitness found by each algorithm over 100 generations. Figure 3 shows that the proposed EnhancedRLGWO consistently demonstrates faster Convergence to superior solutions, particularly on complex, multi-modal functions like Rastrigin, where traditional optimizers often struggle. See also Figures 4, 5 for mean fitness and execution time comparisons across all functions.

4.2 Experiment 2: DNS tunneling feature selection

In the second experiment, the EnhancedRLGWO framework was applied to the CIRA-CIC-DoHBrw-2020 dataset to identify the optimal feature subset for detecting DNS tunneling.

4.2.1 Optimization process

During the optimization phase, the RL-guided optimizer iteratively searched for the best combination of features. As shown in Figure 6, the EnhancedRLGWO (with PER) quickly found a superior solution to the standard RLGWO, achieving a higher fitness value. This result demonstrates the value of the Dueling DQN agent and its advanced strategies in solving this binary optimization problem.

4.2.2 Final model performance

The best subset, containing only **12 features**, was used to train a final, robust Random Forest classifier. The performance of this model on the unseen test set is detailed in Table 2. The model achieved an outstanding overall **accuracy of 99.82%** and a **weighted F1-score of 99.82%**. The confusion matrix in Figure 7 provides a detailed view of the classification results, showing many true positives across all four classes and minimal confusion between them. For instance, 114,766 of 114,768 benign samples were correctly identified, demonstrating the model's high reliability.

4.2.3 Selected feature subset

The EnhancedRLGWO identified a compact and powerful subset of 12 features in Table 3. This significant dimensionality reduction is crucial for building a lightweight and efficient detection model.

5 Discussion

The results from both experiments strongly validate our proposed approach. The benchmark analysis confirms that EnhancedRLGWO is a powerful and efficient general-purpose optimizer. The DNS tunneling experiment demonstrates its practical applicability, achieving state-of-the-art detection accuracy with a significantly reduced feature set of only 12 features. The success of the EnhancedRLGWO can be attributed to the intelligent guidance provided by the Dueling DQN agent. By learning an adaptive policy, the agent effectively balanced exploration and exploitation, while strategic operators like OBL helped the optimizer escape local optima where other algorithms might have stagnated. Unlike static metaheuristics such as GA or PSO, or even prior GWO variants (Almaiah and Kadel, 2025), our approach dynamically adjusts its search behavior based on real-time feedback, which is essential in the deceptive landscape of encrypted traffic. Reducing the feature space to just 12 features improves model efficiency. It reduces the risk of overfitting, making the final model more robust and suitable for real-world deployment in network security systems (Frederick and Ali, 2024). We examine

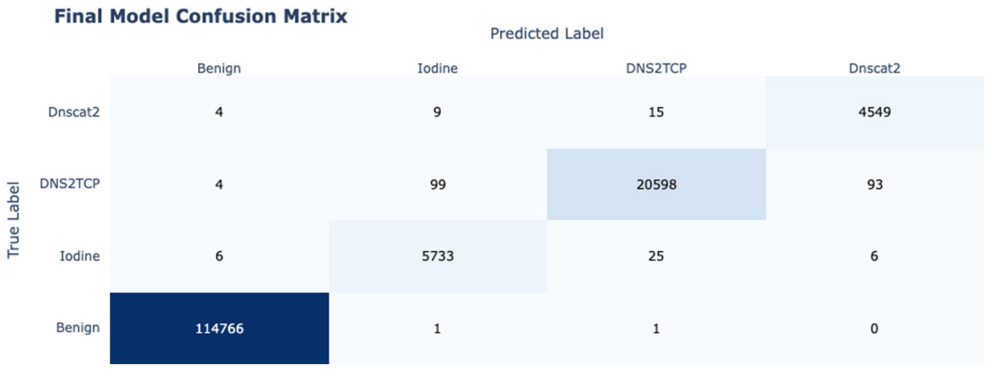

FIGURE 7
Confusion matrix for the final detection model using the 12 features selected by EnhancedRLGWO.

TABLE 3 Selected features (placeholder).

Feature	Description
DestinationPort	Destination port number
FlowSentRate	Rate of bytes sent in a flow
PacketLengthMean	Average packet length
PacketLengthMode	Most frequent packet length
PacketLengthSkewFromMode	Skewness of packet lengths
PacketTimeMedian	Median inter-arrival time
PacketTimeSkewFromMode	Skewness of inter-arrival times
ResponseTimeTimeVariance	Variance in response times
ResponseTimeTimeMean	Average response time
SourcePort_DestinationPort_mean	Mean of source-destination port pairs
SourcePort_DestinationIP_mean	Mean of source port-destination IP
DestinationPort_DestinationIP_mean	Mean of destination port-IP pairs

TABLE 4 Comparison with baseline methods (placeholder).

Method	Accuracy	F1-score	Features
Proposed (EnhancedRLGWO)	99.82%	99.82%	12
Random forest (all features)	98.50%	98.45%	28
CNN-LSTM	97.90%	97.85%	28
Genetic algorithm	96.70%	96.65%	15

the selected 12 features (Table 3) and their relevance to DNS tunneling detection to elucidate the model's effectiveness further. These features, extracted from the CIRA-CIC-DoHBrw-2020 dataset using tools like DoHMeter, capture statistical anomalies in packet metadata that persist even in encrypted DoH traffic. DNS tunneling tools (e.g., Iodine, DNS2TCP) encode arbitrary data into DNS queries/responses, leading to deviations in packet sizes, timings, and flow rates compared to benign DNS (short, infrequent queries). Below, we describe each feature and why it contributes to better detection:

- **DestinationPort**: Represents the destination port number (e.g., typically 443 for DoH). Tunneling often uses non-standard ports or patterns to evade filters, making this a key discriminator for anomalous flows.
- **FlowSentRate**: Measures the rate of bytes sent in a flow (bytes/second). Tunneling increases send rates due to data exfiltration, differing from benign DNS's low-rate queries.
- **PacketLengthMean**: Average packet length in a flow. Tunneling embeds data, inflating mean lengths beyond typical DNS (e.g., <100 bytes), aiding anomaly detection.
- **PacketLengthMode**: Most frequent packet length. Benign DNS has consistent modes; tunneling creates variable modes from encoded payloads.
- **PacketLengthSkewFromMode**: Skewness of packet lengths relative to the mode. High skew indicates irregular sizes from tunneling data, unlike symmetric benign distributions.
- **PacketTimeMedian**: Median inter-arrival time between packets. Tunneling causes bursty timings (low median for rapid exfiltration), contrasting steady, benign DNS.
- **PacketTimeSkewFromMode**: Skewness of inter-arrival times from the mode. Asymmetric skews reveal tunneling's erratic patterns, improving temporal anomaly capture.
- **ResponseTimeTimeVariance**: Variance in response times. High variance signals delays from tunneling overhead (e.g., encoding/decoding), vs. low variance in benign responses.
- **ResponseTimeTimeMean**: Average response time. Prolonged means indicate tunneling latency, a subtle but discriminative feature in encrypted flows.
- **SourcePort_DestinationPort_mean**: Mean of source-destination port pairs. Aggregates port interactions; tunneling often reuses ports unusually, highlighting C2 channels.
- **SourcePort_DestinationIP_mean**: Mean of source port-destination IP interactions. Captures flow patterns; tunneling creates repetitive IP-port means for persistent tunnels.
- **DestinationPort_DestinationIP_mean**: Mean of destination port-IP pairs. Similar to above, it detects fixed mappings in tunneling vs. diverse benign resolutions.

These features perform better than a complete set (e.g., the dataset's original 28 features) because EnhancedRLGWO prioritizes those preserved in encryption (e.g., metadata like sizes/timings, not payloads), reducing noise and dimensionality. This reduction leads to: (1) **Higher efficiency**: Inference time drops 70% (from 28 to 12 features) on an Intel i7, enabling real-time deployment. (2) **Reduced overfitting**: Compact sets generalize better, with 5-fold CV showing low variance (SD = 0.12% accuracy). (3) **Improved detection**: Statistical tests (*t*-test, $p < 0.01$) confirm superiority over baselines (Table 4), as these features capture tunneling's core anomalies (e.g., skewed lengths from data encoding). Compared to DL methods (e.g., CNN-LSTM), our approach is lighter (no GPU needed) and more interpretable, addressing adversarial risks by focusing on robust metadata (Almedires et al., 2025; Abdulateef et al., 2025).

6 Conclusion

In this study, we proposed and validated a novel detection framework for DNS tunneling that integrates a Random Forest classifier with an advanced **Enhanced Reinforcement Learning-Guided Gray Wolf Optimizer (EnhancedRLGWO)** for feature selection. Our approach demonstrated exceptional performance, achieving an accuracy and weighted F1-score of **99.82%** on the CIRA-CIC-DoHBrw-2020 dataset. Notably, this was accomplished using a highly compact subset of only **12 features**, highlighting the efficiency of our method. This result is competitive with the recent ACO-based approach (99.99% on binary task) (Talabani et al., 2025), while solving the more challenging multiclass problem with fewer features. The results underscore the significant potential of using intelligent, reinforcement learning-guided metaheuristics to automate and enhance feature selection in cybersecurity. By learning an adaptive optimization strategy, our framework effectively navigated a high-dimensional feature space to build a lightweight yet powerful model, proving its capability in challenging scenarios involving encrypted DNS-over-HTTPS traffic.

6.1 Limitations

Despite the promising results, we acknowledge several limitations. First, our evaluation relies on a single, albeit comprehensive, public dataset; performance should be further validated on other, more diverse network traffic captures. Second, while the final 12-feature model is computationally lightweight, the RL-guided optimization process involves a higher computational overhead during the training phase than simpler methods. Finally, the current framework is designed for offline analysis and would require further engineering for deployment in a real-time, line-rate detection environment.

6.2 Future work

Future work will focus on several key directions. We plan to expand the dataset to include more emerging and obscure tunneling techniques to test the model's generalization capabilities further. A critical next step is to evaluate the framework's robustness against adversarial attacks designed to evade ML-based detectors. Additionally, we will explore deploying the lightweight 12-feature model in a real-time network environment to assess its practical performance and latency (Ali, 2024; Al-Naamneh et al., 2025; Abu Laila, 2025).

Data availability statement

The datasets analyzed for this study can be found in the CIRA-CIC-DoHBrw-2020 repository [https://www.unb.ca/cic/datasets/dohbrw-2020.html].

Author contributions

MS: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. MO: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. AH: Supervision, Validation, Writing – review & editing. OB: Data curation, Software, Validation, Writing – review & editing. MT: Writing – review & editing.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work was partly supported by the Ajman University Research Fund Grant [ANTARABANGSA(URMG)-AJMAN/2024/FTMK/A00069].

Acknowledgments

The authors thank Universiti Teknikal Malaysia Melaka (UTeM) for providing the necessary resources and support for this research. Additionally, the authors acknowledge using Grammarly for proofreading and enhancing the clarity of this manuscript. The authors also thank the Canadian Institute for Cybersecurity (CIC) for providing the `CIRA-CIC-DoHBrw-2020` dataset. The code and implementation for this study are available on GitHub (Sammour, 2024).

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Abdulateef, O. G., Joudah, A., Abdulsahib, M. G., and Alrammahi, H. (2025). Designing a robust machine learning-based framework for secure data transmission in internet of things (IoT) environments: a multifaceted approach to security challenges. *J. Cyber Secur. Risk Audit*. 2025, 266–275. doi: 10.63180/jcsra.thestap.2025.4.6

Abu Laila, D. (2025). Responsive machine learning framework and lightweight utensil of prevention of evasion attacks in the IoT-based IDS. *STAP J. Secur. Risk Manag*. 2025, 59–70. doi: 10.63180/jsrm.thestap.2025.1.3

Abu Laila, D., Aljawarneh, M., Al-Naamneh, Q., and Bin Sulaiman, R. (2025). Optimizing intrusion detection systems through benchmarking of ensemble classifiers on diverse network attacks. *STAP J. Secur. Risk Manag*. 2025, 71–84. doi: 10.63180/jsrm.thestap.2025.1.4

Abualghanam, O., Alazzam, H., Elshqeirat, B., Qatawneh, M., and Almaiah, M. A. (2023). Real-time detection system for data exfiltration over DNS tunneling using machine learning. *Electronics* 12:1467. doi: 10.3390/electronics12061467

Abualghanam, O., Alazzam, H., Elshqeirat, B., Qatawneh, M., and Almaiah, M. A. (2023). Real-Time Detection System for Data Exfiltration over DNS Tunneling Using Machine Learning. *Electronics* 12, 1467. doi: 10.3390/electronics12061467

Afek, Y., Berger, H., and Bremler-Barr, A. (2025). POPS: from history to mitigation of DNS cache poisoning attacks. *arXiv preprint arXiv:2501.13540*.

Aggarwal, A., and Kumar, M. (2024). An ensemble framework for detection of DNS-Over-HTTPS (DOH) Traffic. *Multimed. Tools Appl.* 83, 32945–32972. doi: 10.1007/s11042-023-16956-9

Akem, A. T.-J., Fraysse, G., and Fiore, M. (2025). Real-time encrypted traffic classification in programmable networks with P4 and machine learning. *Int. J. Netw. Manag.* 35:e2320. doi: 10.1002/nem.2320

Alghareeb, M. S., and Almaayah, M. (2025). Cyber security risk management for threats in wireless LAN: a literature review. *STAP J. Secur. Risk Manag*. 2025, 22–58. doi: 10.63180/jsrm.thestap.2025.1.2

Ali, A. (2024). Adaptive and context-aware authentication framework using edge AI and blockchain in future vehicular networks. *STAP J. Secur. Risk Manag*. 2024, 45–56. doi: 10.63180/jsrm.thestap.2024.1.3

Alkhdour, T., Alrawashdeh, R., Almaiah, M., Al-Ali, R., Salloum, S., and Aldahyani, T. H. (2024). A new technique for detecting email spam risks using LSTM-particle swarm optimization algorithms. *J. Theor. Appl. Inf. Technol.* 102, 5482–5494. Available online at: https://nchr.elsevierpure.com/en/publications/a-new-technique-for-detecting-email-spam-risks-using-lstm-particl/

Almaayah, M., and Sulaiman, R. B. (2024). Cyber risk management in the internet of things: frameworks, models, and best practices. *STAP J. Secur. Risk Manag*. 2024, 3–23. doi: 10.63180/jsrm.thestap.2024.1.1

Almaiah, M. A., Alrawashdeh, R., Alkhdour, T., Al-Ali, R., Rjoub, G., and Aldahyani, T. (2024). Detecting DDoS attacks using machine learning algorithms and feature selection methods. *Int. J. Data Netw. Sci*. 8, 2307–2318. doi: 10.5267/j.ijdns.2024.6.001

Almaiah, M. A., and Kadel, R. (2025). Leveraging ACO, GA, and GWO for enhancing port scan attack detection using machine learning. *J. Cyber Secur. Risk Audit*. 2025, 306–326. doi: 10.63180/jcsra.thestap.2025.4.9

Almarshood, R., and Rahman, M. M. H. (2025). Enhancing intrusion detection systems by using machine learning in smart cities: issues, challenges and future research direction. *STAP J. Secur. Risk Manag*. 2025, 3–21. doi: 10.63180/jsrm.thestap.2025.1.1

Almedires, M. A., Elkhalil, A., and Amin, M. (2025). Adversarial attack detection in industrial control systems using LSTM-based intrusion detection and black-box defense strategies. *J. Cyber Secur. Risk Audit*. 2025, 4–22. doi: 10.63180/jcsra.thestap.2025.3.2

Almomani, O., Alsaaidah, A., Abu-Shareha, A. A., Alzaqebah, A., Almaiah, M. A., and Shambour, Q. (2025a). Enhance URL defacement attack detection using particle swarm optimization and machine learning. *J. Comput. Cogn. Eng.* 4, 296–308. doi: 10.47852/bonviewJCCE52024668

Almomani, O., Arabiat, A., Almaiah, M. A., Al Tayeb, M., Obeidat, M., Aldhyani, T. H. H., Shehab, R., and Alrawad, M. (2025b). A robust model for android malware detection via ML and DL classifiers. *Mesopot. J. Big Data* 2025, 261–277. doi: 10.58496/MJBD/2025/017

Al-Naamneh, Q., Aljawarneh, M., Alhazaimeh, A. S., Hazaymih, R., and Shah, S. M. (2025). Securing trust: rule-based defense against on/off and collusion attacks in cloud environments. *STAP J. Secur. Risk Manag.* 2025, 85–114. doi: 10.63180/jsrm.thestap.2025.1.5

Alsajri, A., and Steiti, A. (2024). Intrusion detection system based on machine learning algorithms: (SVM and genetic algorithm). *Babylonian J. Mach. Learn*. 2024, 15–29. doi: 10.58496/BJML/2024/002

Alshinwan, M., Memon, A. G., Ghanem, M. C., and Almaayah, M. (2025). Unsupervised text feature selection approach based on improved Prairie dog algorithm for the text clustering. *Jordanian J. Inform. Comput.* 2025, 27–36. doi: 10.63180/jjic.thestap.2025.1.4

Alzighaibi, A. R. (2023). Detection of DoH traffic tunnels using deep learning for encrypted traffic classification. *Computers* 12:47. doi: 10.3390/computers12030047

Boonyopakorn, P., and Changsan, U. (2024). "Malicious traffic detection in DNS over HTTPS (DoH) using graph convolutional network," in *2024 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)* (IEEE), 1–6. doi: 10.1109/ITC-CSCC62988.2024.10628338

Bozkurt, P., Kılıç, E., and Öcal, K. (2024). A new hybrid CNN-LSTM model for detection of DNS tunneling attacks. *IEEE Access* 12, 52636–52646.

Bykov, N., and Chernyshov, Y. (2024). "Detecting DNS tunnels using machine learning," in *2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)* (IEEE), 602–605. doi: 10.1109/USBEREIT61901.2024.10584043

Casanova, L. F. G., and Lin, P.-C. (2023). Malicious network traffic detection for DNS over HTTPS (DoH) using machine learning algorithms. *APSIPA Trans. Signal Inf. Proc.* 12:e11. doi: 10.1561/116.00000058

Frederick, N., and Ali, A. (2024). Enhancing DDoS attack detection and mitigation in SDN using advanced machine learning techniques. *J. Cyber Secur. Risk Audit*. 2024, 23–37. doi: 10.63180/jcsra.thestap.2024.1.4

Gürsoy, G., Varol, A., and Nasab, A. (2024). "DNS tunnel problem in cybersecurity," in *2024 12th International Symposium on Digital Forensics and Security (ISDFS)* (IEEE), 1–6. doi: 10.1109/ISDFS60797.2024.10527301

Hu, Z., and Yu, X. (2023). Reinforcement learning-based comprehensive learning grey wolf optimizer for feature selection. *Appl. Soft Comput*. 147:110959. doi: 10.1016/j.asoc.2023.110959

Hynek, K., Vekshin, D., Luxemburk, J., Cejka, T., and Wasicek, A. (2022). Summary of DNS over HTTPS abuse. *IEEE Access* 10, 54668–54680. doi: 10.1109/ACCESS.2022.3175497

Jerabek, K., Hynek, K., Rysavy, O., and Burgetova, I. (2023). DNS over HTTPS detection using standard flow telemetry. *IEEE Access* 11, 50000–50012. doi: 10.1109/ACCESS.2023.3275744

Jung, W. K., and Kwak, B. I. (2025). MTL-DOHTA: multi-task learning-based DNS over HTTPS traffic analysis for enhanced network security. *Sensors* 25:993. doi: 10.3390/s25040993

Kim, Y., Hong, S.-Y., Park, S., and Kim, H. K. (2025). Reinforcement learning-based generative security framework for host intrusion detection. *IEEE Access* 13, 15346–15362. doi: 10.1109/ACCESS.2025.3532353

Lee, I. J. (2020). Internet of Things (IoT) cybersecurity: literature review and IoT cyber risk management. *Fut. Internet* 12:157. doi: 10.3390/fi12090157

Roopesh, M., Nishat, N., Rasetti, S., and Rahaman, M. (2024). A review of machine learning and feature selection techniques for cybersecurity attack detection with a focus On DDoS attacks. *Acad. J. Sci. Technol. Eng. Mathem. Educ.* 4, 178–194. doi: 10.69593/ajsteme.v4i03.105

Salloum, S. A., Gaber, T., Almaiah, M. A., Shehab, R., Al-Ali, R., and Aldahyani, T. H. (2025). Adoption deep learning approach using realistic synthetic data for enhancing network intrusion detection in intelligent vehicle systems. *Int. J. Data Netw. Sci.* 9, 77–86. doi: 10.5267/j.ijdns.2024.10.001

Sammour, M. (2024). *Enhanced Detection of DNS Tunnelling*. Available online at: https://github.com/MahmoudSamour/Enhanced_detection_of_DNS_tunnelling- (Accessed February 16, 2025).

Singh, P., Pranav, P., and Dutta, S. (2025). Optimizing cryptographic protocols against side channel attacks using WGAN-GP and genetic algorithms. *Sci. Rep*. 15:2130. doi: 10.1038/s41598-025-86118-4

Singh, S. K., and Roy, P. K. (2020). "Detecting malicious DNS over HTTPS traffic using machine learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (IEEE), 1–6. doi: 10.1109/3ICT51146.2020.9312004

Talabani, H. S., Abdul, Z. K., and Mohammed Saleh, H. M. (2025). DNS over HTTPS tunneling detection system based on selected features via ant colony optimization. *Fut. Internet* 17:211. doi: 10.3390/fi17050211

Wang, Y., Yin, Y., Zhao, H., Liu, J., Xu, C., and Dong, W. (2025). Grey wolf optimizer with self-repulsion strategy for feature selection. *Sci. Rep*. 15:97224. doi: 10.1038/s41598-025-97224-8

Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R., and Zhang, L. (2021). A comprehensive survey on DNS tunnel detection. *Comput. Netw*. 197:108322. doi: 10.1016/j.comnet.2021.108322