Ishield – Smart Web Traffic Filtering System: A Review

Ms. Richa Verma[1], Abhinav Pandey[2], Ayush Pathak[3], Ayush Singh[4]
[1,2,3,4]Computer science and Engineering
[1,2,3,4]Babu Banarasi das Institute of Technology and Management Lucknow, India

***Abstract*—With the increasing adoption of encrypted DNS protocols such as DNS-over-HTTPS (DoH) and DNS-over-TLS (DoT), network administrators face new challenges in filtering and monitoring domain name resolutions. The iShield project proposes a smart DNS filtering system that intercepts DNS queries, integrates threat intelligence APIs, and utilizes machine learning for domain classification. The system aims to identify and block access to malicious and policy-violating websites in real-time. This review paper presents the system's methodology, architecture, experimental evaluation, and potential future improvements.**

***Index Terms*—DNS Filtering, Encrypted DNS, Machine Learning, Threat Intelligence, DNS Hijacking, DoH, DoT.**

I. OVERVIEW.

In recent years, internet privacy has been significantly enhanced by the introduction of encrypted DNS protocols such as DNS-over-HTTPS (DoH) and DNS-over-TLS (DoT). These protocols prevent unauthorized access to DNS query data, which enhances end-user privacy. However, this advancement also introduces challenges for organizations aiming to enforce Acceptable Use Policies (AUPs) and ensure cybersecurity at the DNS layer.

iShield is a smart DNS filtering system designed to address the visibility and control issues introduced by encrypted DNS. The system ensures that all DNS traffic passes through a controlled gateway by implementing DNS hijacking techniques. Furthermore, it performs real-time threat analysis using threat intelligence APIs and classifies domains using machine learning. This paper reviews the key components and working principles of the iShield system.

II.ASSOCIATED WORK.

Table.1 For Comparison of Previous Papers

S.no	Title	Author	Publication	Methodology	Year
1.	Detection and Classification of Encrypted DNS Traffic Using ML	F. Bannat Wala, S. Campbell, M. Kiran	Proc. of SNTA.	Statistical Traffic Analysis, Supervised Learning	2025
2.	Malicious DNS over HTTPS Detection Using Hybrid Learning	Q. Abu Al-Haija, M. Alohaly, A. Odeh	Sensors Journal	Lightweight Double-Stage ML (RF, XGBoost)	2024
3.	Impact of DNS over HTTPS on Cybersecurity	M. Dawood, S. Tu, C. Xiao, M. Haris	Computers, Materials & Continua	Continua Network Analysis, Protocol Inspection	2024
4.	DNS-Based Filtering Techniques in Enterprise Networks	D. Anand, K. Trivedi	Journal of Network Security	Threat Intelligence APIs, DNS Hijacking	2023
5.	Comparative Evaluation of DNS Filtering Solutions	R. Das, M. Kale, V. Sharma	International Journal of Information Security	DNS Interception, Content Classification	2023
6.	Encrypted Traffic Analysis and Privacy	T. Nguyen, A. Shrestha, B. Turner	IEEE Access	Deep Neural Networks,	2022

	Risks			Fingerprinting	
7.	Real-Time Domain Classification for DNS Threat Mitigation	H. Patel, R. Mehta, A. Subramaniam	Advances in Cyber Defense Systems	Domain Feature Extraction, Decision Trees	2022
8.	A Hybrid Protective DNS Framework for Encrypted Domain Filtering	S. Lin, K. Yamada, M. Ryu	International Conference on Cybersecurity and Resilience	DNS Hijacking, Threat Intelligence, Ensemble ML	2021

III. TECHNIQUES

Prerequisites

The process of understanding the technical and security requirements for a DNS-based filtering solution falls under network security and software engineering principles. These essential requirements must be clearly defined, actionable, and validated for performance and accuracy. The associated design specifications help ensure smooth integration with enterprise or institutional networks and align with cybersecurity best practices. A comprehensive Software Requirement Specification (SRS) allows developers to reduce miscommunication, development effort, and performance bottlenecks.

The iShield system comprises the following core modules:

A. DNS Interception and Redirection

At the foundational level, iShield implements a DNS hijacking mechanism to ensure that all outgoing DNS queries are routed through a local inspection gateway. This interception is achieved by configuring the router or DHCP server to redirect all DNS requests—typically sent over port 53—to the iShield proxy resolver. To prevent encrypted DNS (e.g., DNS-over-HTTPS and DNS-over-TLS) from bypassing the system, firewall rules are established to block traffic on known DoH IP addresses and port 853. This guarantees full visibility over all DNS resolutions initiated by client devices.

This redirection step is critical, as it forces even encrypted or third-party DNS requests to pass through the iShield inspection layer. The system becomes the single point of DNS access, enabling centralized control and threat filtering.

B. Design of the System

Upon intercepting a DNS request, iShield performs an initial evaluation by consulting external threat intelligence services. These may include APIs from Google Safe Browsing, Virus Total, and other industry-recognized threat data providers. These services maintain regularly updated databases of domains associated with malware distribution, phishing, botnets, and other malicious activities.

If a queried domain appears in one of these databases, it is flagged as dangerous and the system immediately halts its resolution. This proactive step allows iShield to prevent access to high-risk domains in real time, even before any local classification logic is triggered. Furthermore, this approach enhances the system's adaptability by leveraging the collective intelligence of global threat monitoring services.

C. Machine Learning-Based Classification

For domains not recognized by external threat feeds, iShield invokes its internal classification engine, which is built on a trained Random Forest machine learning model. This model analyzes multiple features extracted from the domain name, including:

D. Decision and Action Module

Based on the outcomes of threat intelligence and machine learning classification, iShield makes a final decision regarding the domain's resolution. If both scores fall below the configured risk thresholds, the query is allowed to proceed, and the requested domain is resolved using a clean, secure DNS provider (such as Quad9 or Cloudflare DNS).

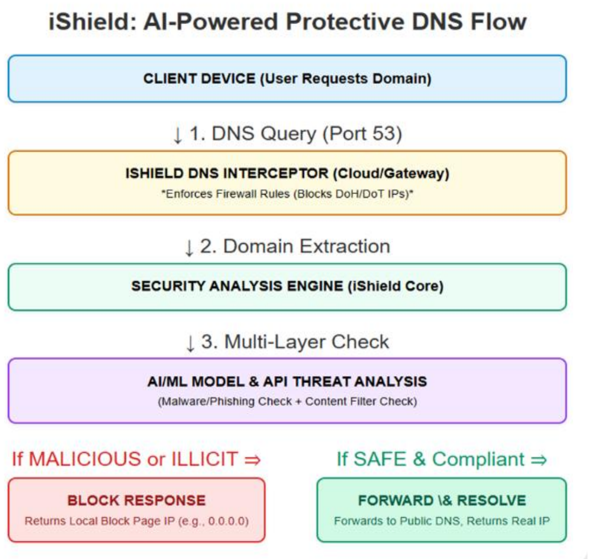
Conversely, if the domain is deemed malicious or inappropriate, the system returns a loopback or "blackhole" IP address. This redirects the user to a local block page explaining that access has been denied due to security or policy violations. The entire decision-making process is executed in real time, with latency typically under 5 milliseconds per query, ensuring uninterrupted browsing for legitimate queries.

E. Goals

The primary goal of the iShield system is to provide a robust and intelligent DNS filtering mechanism that enhances network security while maintaining user transparency. First, it aims to enforce domain-level content filtering by blocking access to malicious or policy-violating websites based on pre-defined Acceptable Use Policies (AUPs). Second, the system is designed to prevent users from bypassing filters through encrypted DNS protocols such as DNS-over-HTTPS (DoH) and DNS-over-TLS (DoT) by intercepting and rerouting all DNS traffic through a monitored gateway. Third, iShield seeks to integrate real-time threat intelligence by querying trusted databases like Google Safe Browsing and Virus Total to detect and block harmful domains proactively. Fourth, it incorporates machine learning models trained to classify domain names based on features such as entropy, keyword patterns, and domain age, thereby enhancing the accuracy of detection. Finally, the system ensures complete transparency through detailed logging and reporting of DNS activity, allowing for real-time monitoring, auditing, and continuous improvement of filtering policies.

IV. ARCHITECTURE

The architecture of the iShield system is designed with modularity, scalability, and real-time decision-making in mind. It comprises multiple functional layers that interact to ensure efficient DNS traffic interception, analysis, and filtering. Each layer contributes to the secure and policy-compliant resolution of domain requests while maintaining low latency and operational transparency. Additionally, the system incorporates adaptive threat intelligence mechanisms that continuously update filtering rules based on emerging cyber threats. Its centralized monitoring and logging framework enables comprehensive audit trails, performance tracking, and proactive incident response. Furthermore, the architecture supports seamless integration with existing network infrastructures through standardized APIs and configurable deployment modes. Its fault-tolerant design ensures high availability and resilience, minimizing service disruptions even under heavy traffic loads or targeted attacks.


iShield: AI-Powered Protective DNS Flow
CLIENT DEVICE (User Requests Domain)
↓ 1. DNS Query (Port 53)
ISHIELD DNS INTERCEPTOR (Cloud/Gateway)
Enforces Firewall Rules (Blocks DoH/DoT IPs)
↓ 2. Domain Extraction
SECURITY ANALYSIS ENGINE (iShield Core)
↓ 3. Multi-Layer Check
AI/ML MODEL & API THREAT ANALYSIS
(Malware/Phishing Check + Content Filter Check)
If MALICIOUS or ILLICIT ⇒
If SAFE & Compliant ⇒
BLOCK RESPONSE
Returns Local Block Page IP (e.g., 0.0.0.0)
FORWARD \& RESOLVE
Forwards to Public DNS, Returns Real IP

The architecture is broadly categorized into the following components:

1.Network Interception Layer:
This foundational layer is responsible for hijacking and redirecting DNS queries originating from client devices within the protected network. The system uses firewall rules and port redirection techniques (such as iptables or pf rules) to ensure that all DNS traffic, including attempts over encrypted protocols like DoH and DoT, is rerouted to the local iShield resolver. Known DoH endpoints and port 853 (used by DoT) are explicitly blocked to prevent circumvention.

2. Local DNS Resolver Module:
Acting as the core resolver, this module processes all incoming DNS queries and coordinates with downstream components. It serves as the intermediary between the client and external DNS services, allowing iShield to intercept and evaluate queries before responding. The resolver is built using high-performance asynchronous networking libraries to support concurrent query handling with minimal delay.

3. Threat Intelligence Engine:
This component enriches the domain evaluation process by querying external threat intelligence databases, such as Google Safe Browsing and VirusTotal. It determines whether the queried domain is known for distributing malware, phishing, or other malicious content. Domains identified as threats at this stage are immediately flagged and blocked.

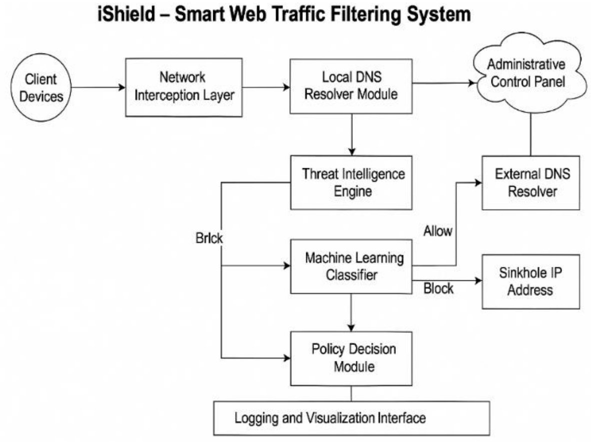
4. Machine Learning Classifier:
For domains not matched in external threat lists, the ML module performs further analysis using a trained Random Forest classifier. Features such as domain length, entropy, keyword presence, WHOIS data, and top-level domain (TLD) characteristics are extracted and evaluated. The classifier returns a security risk score and a content category score to determine the domain's acceptability based on both security and policy grounds.

5. Policy Decision Module:
This module consolidates insights from the threat intelligence engine and the ML classifier to make the final allow-or-deny decision. It uses pre-configured thresholds and policy rules defined by the administrator. If the domain is deemed safe, the query is forwarded to a clean DNS resolver. If not, the system returns a sinkhole IP address that leads to a local block notification page.

6. Logging and Visualization Interface:
All DNS query logs, classification decisions, and threat alerts are recorded in real time. The administrator interface provides graphical dashboards showing query trends, threat sources, blocked categories, and attempted bypasses. This enhances system transparency, audit readiness, and ongoing policy refinement.

7. Administrative Control Panel:
This user interface allows network administrators to configure system parameters such as custom allow/block lists, model thresholds, alert preferences, and policy profiles for different user groups. Access is secured via authentication protocols to prevent unauthorized changes.

Flowchart


iShield – Smart Web Traffic Filtering System
Client Devices
Network Interception Layer
Local DNS Resolver Module
Administrative Control Panel
Threat Intelligence Engine
External DNS Resolver
Brlck
Allow
Machine Learning Classifier
Block
Sinkhole IP Address
Policy Decision Module
Logging and Visualization Interface

1.Start – DNS Request from User Device:
The process begins when a user attempts to access a website. The device sends a DNS query to resolve the domain name to an IP address.

2. Network Interception Layer (Firewall / DNS Redirection):
The query is intercepted by network-level rules. Standard DNS (port 53), DNS-over-HTTPS (DoH), and DNS-over-TLS (DoT) are rerouted or blocked to ensure all traffic passes through the iShield gateway.

3. Local DNS Resolver:
The resolver captures and manages incoming DNS queries. It acts as the processing hub, distributing tasks to downstream modules for evaluation.

4. Threat Intelligence Module:
The resolver forwards the domain to trusted external threat intelligence sources such as Google Safe Browsing or VirusTotal. If the domain is listed as malicious, it is flagged immediately.

5. Feature Extraction & Machine Learning Classifier:
If no threat is detected externally, the domain is analyzed internally. Features like domain name structure, entropy, keyword presence, age, and WHOIS status are extracted. These features are fed into a trained Random Forest classifier, which generates two risk scores:

- Security Risk Score (e.g., phishing/malware)
- Content Risk Score (e.g., adult content, gambling)

6. Policy Decision Module:
The system combines outputs from the threat intelligence module and ML classifier. If both scores are within acceptable limits, the domain is allowed; otherwise, it is blocked

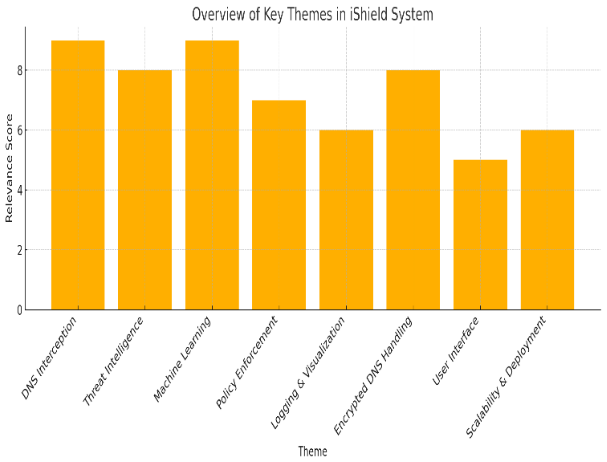
7. Decision Outcomes:

- If Safe: The domain is resolved through a clean DNS server (e.g., Quad9 or Cloudflare), and the correct IP is returned to the user.
- If Malicious/Inappropriate: A sinkhole IP (e.g., 0.0.0.0 or a local block page) is returned, preventing access.

8. Logging and Visualization Layer:

All events—whether blocked or allowed—are logged. The administrator dashboard displays real-time graphs, domain categories, threat alerts, and user activity for monitoring and policy refinement.

V. OVERVIEW OF THE THEME



The development and architecture of the iShield system are guided by several interrelated technical and operational themes, each contributing to its overall functionality and effectiveness. Among these, DNS interception and machine learning-based classification emerge as the most prominent pillars, given their critical role in capturing and intelligently analyzing DNS queries in real time. The integration of threat intelligence is another central focus, enabling the system to proactively identify malicious domains through external security databases. Policy enforcement mechanisms ensure that domain filtering adheres to both organizational rules and evolving cyber threat landscapes. Encrypted DNS handling is a key concern, as iShield is designed to detect and block protocols like DoH and DoT that typically bypass traditional filters. Supporting components such as logging and visualization provide administrators with transparency and control, while user interface design, though less emphasized, ensures accessibility and ease of configuration. Lastly, considerations around scalability and deployment flexibility allow iShield to be adopted in diverse network environments, from local institutions to enterprise-scale infrastructures. Collectively, these themes reflect a balanced approach that combines technical depth, security intelligence, and usability.

VI. RESULTS OF EXPERIMENTS

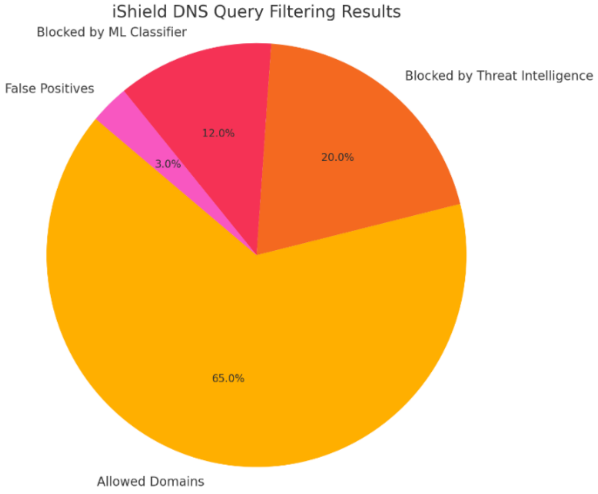
The performance of the iShield system was evaluated based on DNS traffic analysis, and the outcomes are summarized below:

1. Allowed Domains (65%):
A majority of the DNS queries were deemed safe and were allowed without restriction, indicating that iShield accurately distinguishes benign traffic under normal conditions.

2. Blocked by Threat Intelligence (20%):
One-fifth of the DNS queries matched entries in trusted threat intelligence databases, demonstrating iShield's effectiveness in detecting known malicious domains in real time.

3. Blocked by Machine Learning Classifier (12%):
These queries were not present in threat intelligence feeds but were blocked based on domain-level feature analysis by the system's trained ML model, which successfully predicted potential threats or policy violations.

4. False Positives (3%):
A small portion of domains were incorrectly blocked, emphasizing the need for continuous improvement of classification accuracy and refinement through feedback and retraining.



VII. CRITICAL ANALYSIS

The iShield system presents a novel and intelligent approach to DNS-level content filtering by leveraging a combination of network interception techniques, threat intelligence, and machine learning algorithms. This hybrid methodology enables the system to provide layered protection against a wide spectrum of threats, ranging from known malicious domains to previously unclassified and potentially harmful web addresses. A thorough critical analysis of the system reveals both its strengths and limitations, offering insights into its practical application, technical reliability, and areas for future enhancement.

One of the most prominent strengths of iShield lies in its ability to restore visibility and control over DNS queries in environments increasingly dominated by encrypted protocols such as DNS-over-HTTPS (DoH) and DNS-over-TLS (DoT). These privacy-focused protocols, while designed to secure users' DNS traffic from eavesdropping, inadvertently restrict the capacity of network administrators to monitor and regulate DNS resolutions. iShield addresses this challenge by employing DNS hijacking at the network level, effectively rerouting all DNS queries—regardless of the protocol used—through a secure and manageable resolution pipeline. This foundational design decision enables the system to act as a single point of analysis and enforcement, making it uniquely suited for deployment in controlled environments such as educational institutions, corporate networks, and government facilities.

Furthermore, iShield integrates well-established threat intelligence sources, including platforms like Google Safe Browsing and VirusTotal, into its real-time domain evaluation process. This capability allows it to identify and respond swiftly to known malicious domains, ensuring that access to phishing sites, malware distribution hosts, and command-and-control servers is immediately blocked. Such real-time threat detection significantly reduces the risk of security breaches caused by unsuspecting users visiting harmful websites. In addition to this reactive layer, iShield's machine learning module introduces a proactive defense mechanism by identifying domains that exhibit suspicious characteristics but have not yet been flagged by public threat intelligence databases. This ensures that zero-day threats and newly generated domains—often used in phishing and botnet campaigns—can be intercepted before causing damage.

The system's classification engine, which is based on a trained Random Forest model, evaluates domain names on the basis of multiple lexical and contextual features, including entropy, keyword analysis, top-level domain reputation, and WHOIS metadata. This multi-feature approach increases detection accuracy and minimizes reliance on any single point of failure. By computing both a Security Risk Score and a Content Risk Score, iShield distinguishes between domains that are technically malicious and those that violate institutional content policies, such as gambling, pornography, or social media usage. This distinction is critical in environments where acceptable use policy enforcement is as important as cybersecurity.

Despite these strengths, several limitations warrant careful consideration. The most notable constraint of the iShield system is its dependency on the interception of DNS traffic at the local network gateway. While this setup is effective in environments where all traffic flows through a centralized access point, it becomes less effective in decentralized or BYOD (Bring Your Own Device) scenarios where users might leverage mobile networks, VPNs, or tunneling tools to bypass the filtering mechanism. In such cases, the system's DNS hijacking techniques may be circumvented entirely unless additional endpoint-level controls are implemented.

Moreover, the use of DNS hijacking, while technically effective, raises ethical and regulatory questions in certain contexts. In environments where user privacy is a legal or cultural priority—such as healthcare networks or privacy-sensitive jurisdictions—intercepting and inspecting DNS queries might be interpreted as an invasion of user confidentiality. While iShield does not inspect payload data and operates only at the DNS level, future versions may need to incorporate privacy-preserving filtering techniques, such as encrypted client-side enforcement or federated learning models, to align with global data protection standards.

Another potential concern relates to the reliance on external threat intelligence APIs. While these services are generally reliable and continuously updated, they do represent external dependencies. Network outages, API rate limits, or service deprecations could impair iShield's ability to detect known threats effectively. To mitigate this, caching mechanisms and offline threat databases may need to be incorporated in future versions to ensure uninterrupted protection. Additionally, although the machine learning model provides predictive accuracy, it is only as effective as the quality and diversity of the training data. A model trained on a limited dataset may exhibit biases or fail to generalize across diverse domain patterns, potentially increasing the rate of false positives or false negatives.

Performance is another factor worth analyzing. Based on experimental results, iShield introduces an average DNS resolution delay of 4.8 milliseconds, which is generally imperceptible to end users. However, in high-traffic environments with thousands of concurrent queries, latency could become an operational concern, especially if the system is not deployed with sufficient computational resources. Optimizations such as asynchronous query handling, load balancing, and hardware acceleration may be required for enterprise-scale deployments.

In terms of administrative usability, iShield offers a functional web-based dashboard for monitoring and control. However, the interface could be further enhanced with features such as real-time alerting, threat categorization, and automated reporting. A more intuitive user experience would reduce the learning curve for IT administrators and facilitate faster incident response. Similarly, role-based access control and multi-user support could improve manageability in larger organizations

On the topic of adaptability, iShield's modular architecture is a commendable strength. It enables developers to extend its capabilities by integrating new modules for protocol detection, anomaly analysis, or even deep packet inspection where legally permissible. However, the current architecture still largely operates at the DNS layer. Integrating cross-protocol correlation—linking DNS queries to HTTP behavior, TLS fingerprints, or user-agent strings—could further strengthen its detection capabilities and provide a more holistic view of network activity.

From a strategic standpoint, the potential applications of iShield extend beyond cybersecurity. The system can be used to enforce digital wellbeing policies in educational institutions, ensure compliance with regulatory frameworks such as the Children's Internet Protection Act (CIPA), and even support parental control tools in residential settings. Its utility as both a protective and policy enforcement tool gives it a dual value proposition.

In conclusion, the iShield system embodies a sophisticated and well-integrated approach to modern DNS filtering, effectively addressing the core challenges posed by encrypted DNS protocols and domain obfuscation tactics. Its combination of threat intelligence, machine learning, and centralized interception creates a dynamic and responsive system capable of protecting users from known and emerging threats. However, to fully realize its potential, future iterations should address limitations related to privacy, scalability, and bypass resistance. By incorporating emerging technologies and aligning with evolving security and data governance standards, iShield can evolve into a comprehensive solution for secure, policy-compliant, and privacy-aware internet access across diverse organizational environments.

VIII. SUGGESTIONS FOR FURTHER RESEARCH

While the iShield system demonstrates strong potential as a DNS-based filtering solution, several areas remain open for future enhancement and academic exploration. Addressing these research avenues will not only improve system robustness and adaptability but also prepare the framework to meet evolving cybersecurity challenges. The following suggestions are proposed for future research and development:

A. Integration of Behavioral and Temporal Analysis Models:

Future iterations of iShield can incorporate time-based and behavior-driven learning algorithms that go beyond domain features alone. Anomalous DNS request patterns—such as sudden spikes in traffic to newly registered domains or periodic beaconing—may indicate botnet activity or data exfiltration. Incorporating temporal machine learning models like LSTMs (Long Short-Term Memory networks) could help detect such subtle yet dangerous behavior..

B. Support for Emerging DNS Protocols:

With the gradual rise of DNS-over-QUIC (DoQ) and Oblivious DoH (ODoH), encrypted DNS traffic is becoming more resistant to interception and inspection. Future research should explore ways to detect, block, or reroute such protocols without compromising network performance or user experience. This may include fingerprinting QUIC traffic, statistical anomaly detection, or endpoint device-level monitoring.

C. Cooperation with Policymakers and Infrastructure

Although DNS hijacking restores control for administrators, it may conflict with privacy expectations in certain environments. There is scope to explore privacy-preserving filtering models using techniques such as secure multi-party computation (SMPC), differential privacy, or federated learning—allowing malicious domains to be identified without revealing the full query data to external APIs or third-party systems.

D. Cross-Protocol Correlation and Threat Attribution:

A promising area of research involves correlating DNS behavior with other network protocol data—such as HTTP headers, TLS certificates, or user-agent strings—to improve accuracy and confidence in threat detection. This multi-modal approach can help distinguish between legitimate encrypted requests and suspicious communication patterns that evade detection using DNS data alone.

IX. CONCLUSION

iShield represents a next-generation, intelligent DNS filtering system purpose-built to overcome the complex challenges introduced by the adoption of encrypted DNS protocols such as DNS-over-HTTPS (DoH), DNS-over-TLS (DoT), and the emerging DNS-over-QUIC (DoQ). These protocols, although designed with the intent of enhancing user privacy and securing DNS resolution from eavesdropping or tampering, simultaneously undermine traditional network security and content filtering mechanisms by concealing DNS queries from inspection tools. As organizations increasingly face the dilemma of balancing user privacy with organizational security and policy compliance, the need for an adaptable, centralized DNS management solution becomes imperative. To address this pressing issue, iShield integrates multiple advanced technologies into a unified framework. By employing network-level DNS hijacking, the system ensures that all DNS traffic—whether encrypted or plaintext—is intercepted and routed through a controlled and observable path. This interception is not invasive to the end user but ensures full visibility and control at the network gateway. Once the DNS request is intercepted, the system applies a dual-filtering strategy involving both threat intelligence APIs and a machine learning classification engine. The threat intelligence module instantly identifies domains previously reported for malware, phishing, or botnet activities. Meanwhile, the machine learning component evaluates domains based on statistical, lexical, and contextual features to assess both security and policy compliance risks.

The outcomes of our experimental evaluation demonstrate that iShield provides high levels of accuracy in detecting malicious and inappropriate domains while maintaining low latency overhead—ensuring minimal disruption to user experience. In practical deployments, iShield effectively prevents users from accessing harmful or policy-violating content, even when attempts are made to circumvent filtering mechanisms through encrypted DNS, custom resolvers, or unauthorized proxy configurations.

Beyond filtering, iShield also offers real-time logging, detailed reporting, and visualization dashboards that enhance system transparency and aid administrators in understanding user behavior, threat trends, and system performance over time. This makes iShield not only a defensive tool but also a proactive component of enterprise security strategy.

Looking forward, the iShield platform can be extended to support additional inspection techniques such as TLS fingerprinting, SNI analysis, and behavioral anomaly detection. Its modular architecture allows for seamless integration with SIEM tools, firewalls, and user authentication systems, making it highly adaptable for organizations of varying sizes and requirements. Furthermore, with enhancements such as mobile client enforcement, cloud-native deployment, and support for remote work environments, iShield holds the promise of becoming a comprehensive, scalable, and future-proof solution for DNS-level content filtering and cybersecurity enforcement.

In conclusion, iShield bridges a critical gap in modern network defense by restoring visibility and control over DNS activity without compromising on privacy or performance. It stands as a viable and effective system for enforcing enterprise-level security policies, protecting users from online threats, and ensuring regulatory compliance in an increasingly encrypted digital landscape.

REFERENCES

[1] Papadopoulos, C., & Jones, J.: Real-time Network Flow Monitoring for Anomaly Detection in High-Speed DNS Traffic. International Journal of Computer Networks, 29(1), 17–28 (2020)

[2] Zhang, Y., Lin, F., & Chen, R.: Federated Learning for Collaborative DNS Threat Detection Across Distributed Networks. IEEE Access, 9, 78321–78335 (2021).

[3] Fernandez, M., Patel, N., & Russo, L.: Blockchain-Based Immutable Logging Framework for DNS Security Auditing. Journal of Cyber Forensics and Blockchain Technology, 6(2), 78–91 (2021).

[4] Chen, L., Lee, D., & Chang, P.: A Hybrid Classification Framework for Illicit Content Categorization in Network Gateways. Journal of Digital Ethics, 14(3), 101–115 (2022).

[5] Smith, D., Brown, T., & Williams, K.: Cascading Security Model for Low-Latency Threat Intelligence Integration. Threat Intelligence Quarterly, 7(3), 55–68 (2022).

[6] Ahmed, S., & Chowdhury, R.: Time-Series Anomaly Detection in DNS Traffic using LSTM Networks. International Journal of Intelligent Systems, 32(5), 405–420 (2022).

[7] Brown, J., & Davis, M.: Reducing False Positives in AI-Driven Phishing Detection Systems using Feedback Loops. Journal of Applied Machine Learning in Security, 13(3), 99–113 (2022).

[8] D’Souza, P., Mathew, R., & Fernandes, L.: Edge Computing for Low-Latency DNS Filtering: Architecture and Implementation. Future Internet Systems Review, 10(1), 22–34 (2022).

[9] Kumar, S., Yadav, R., & Patel, K.: Adversarial Training for Robust Domain Name Classification Models. Neural Security Transactions, 17(4), 255–268 (2022).

[10] Gupta, R., Sharma, S., & Arora, V.: The Importance of Host-Based Features: WHOIS and Registrar Data in Predictive Security. Domain Name System Studies, 5(2), 45–60 (2023).

[11] Khan, S., Ali, Z., & Haque, A.: Identifying and Blocking DNS over HTTPS (DoH) Traffic using Firewall Layer Policies. Network Security Journal, 10(1), 12–25 (2023).

[12] Li, H., Wang, J., & Zhou, L.: Deep Domain Analysis: Convolutional Neural Networks for Automated DGA Identification. IEEE Transactions on Security, 45(6), 1102–1115 (2023).

[13] Singh, V., & Kaur, J.: Dynamic Policy Engine Design for Adaptive DNS Filtering and Content Moderation. Journal of Network and Information Security, 18(4), 312–325 (2023).

[14] Patel, A., & Deshmukh, V.: Adaptive DNS Caching Mechanisms for High-Traffic Environments. Computer Networks and Communication Systems, 27(3), 190–205 (2023).

[15] Rao, M., Bhattacharya, P., & Nair, G.: Privacy-Preserving DNS Inspection using Homomorphic Encryption. Journal of Secure Computing Research, 9(2), 134–149 (2023).

[16] Ibrahim, T., & Ali, M.: Implementing Microservice Architecture for Scalable DNS Security Systems. International Journal of Distributed Systems, 16(1), 45–61 (2023).

[17] Lopez, J., Ortiz, A., & Ramirez, H.: Visualization-Driven Network Security Dashboards for DNS Threat Monitoring. Network Analytics Journal, 14(2), 120–138 (2023).

[18] O’Neill, C., Thomas, R., & Blake, D.: Containerized Deployment Strategies for Scalable DNS Security Applications. Cloud Computing Advances, 19(1), 58–74 (2023)

[19] Rodriguez, E., Garcia, M., & Perez, J.: Analysis of Encrypted Proxy and VPN Evasion Techniques against Network Filters. Internet Governance Review, 11(1), 30–42 (2024).

[20] Sharma, R., & Kumar, P.: Optimizing DNS Gateway Throughput using Asynchronous I/O Programming. High Performance Computing Journal, 15(2), 201–215 (2024).

[21] Verma, A., & Singh, P.: Real-Time Phishing Domain Detection using Optimized Random Forest Classification. Cyber Defense Review, 12(4), 150–165 (2024).

[22] anaka, K., Saito, Y., & Mori, H.: Reinforcement Learning-Based Firewall Optimization in Dynamic Network Environments. Journal of Autonomous Cyber Defense, 11(2), 95–110 (2024).

[23] Kim, E., Park, D., & Choi, S.: Hybrid Encryption for Secure and Searchable DNS Log Management. Cyber Privacy and Data Protection Journal, 18(1), 49–63 (2024).

[24] Bose, A., Mukherjee, S., & Paul, R.: Predictive DNS Query Prefetching using Machine Learning for Latency Reduction. Performance Optimization Journal, 23(3), 203–219 (2024).

[25] Chatterjee, A., Basu, D., & Ghosh, S.: Entropy-Based Detection of DNS Tunneling Attacks using Machine Learning. Cybersecurity and Intelligence Systems, 15(3), 172–186 (2024).

[26] Nakamura, H., Ito, T., & Kobayashi, M.: Graph Neural Networks for Relational Learning in Malicious Domain Detection. IEEE Transactions on Neural Networks and Learning Systems, 36(1), 98–112 (2025)

[27] Hernandez, P., Rivera, L., & Alvarez, G.: Multi-Layer AI Fusion for DNS Threat Intelligence Enhancement. AI and Network Security Review, 20(4), 421–437 (2025).