Data Engineering and Applications

Research Paper

Physics-Informed Diffusion Policy Transformer Model for DNS Tunnel Traffic Discovery

Stefan Jovanović[1, *], Miloš Stojanović[1] and Zoran Mladenović[1]

[1] Faculty of Informatics and Computing, Singidunum University, Belgrade, 11070, Serbia
*Corresponding author: stefan.jov@singidunum.ac.rs

Abstract. This study develops a physics-informed Diffusion Policy Transformer for DNS tunnel traffic discovery in encrypted, mixed, and low-rate network scenarios. The method addresses the difficulty of identifying covert DNS behavior when single query records contain only weak evidence. Query length, label entropy, answer code, time interval, packet direction, host-domain persistence, resolver path, TTL variation, NXDOMAIN ratio, and domain-structure characteristics are encoded as protocol-physical descriptors. In this context, physical information refers to DNS protocol legality and operational traffic constraints rather than mechanical laws. A Transformer encoder learns long-range host-domain session dependencies, and a conditional diffusion module iteratively denoises the latent tunnel-risk state before adaptive threshold calibration produces analyst-facing warning grades. Experiments are conducted on a simulated enterprise DNS monitoring dataset containing 210,000 DNS query records, 31,600 host-domain sessions, and 36 protocol-physical characteristics. Compared with rule filtering, random forest, BiLSTM, Transformer, and non-physical diffusion baselines, the proposed model raises tunnel F1 from 0.884 to 0.932, improves low-rate tunnel recall from 81.6% to 90.7%, and reduces false alarms by 23.4%. Ablation results indicate that protocol-constraint encoding and diffusion calibration are the main sources of robustness. The study remains limited by its simulated data and should be validated with privacy-preserving enterprise DNS logs, multi-resolver deployments, and adversarial tunnel tools.

Keywords: *DNS Tunnel Discovery, Diffusion Policy Transformer, Protocol-Physical Feature Encoding, Encrypted Traffic Analysis, Low-Rate Covert Channel*

Received on 28 March 2026, Accepted on 29 June 2026, Published on 13 July 2026

Introduction

For practically all corporate networks, DNS is an indispensable service. The same availability that makes DNS necessary for routine operation also makes it attractive for covert communication. DNS tunnelling can conceal command transmission, data exfiltration, malware callbacks, and policy-bypass traffic inside domain-name queries and responses. Such traffic is rarely a simple mass of obvious malicious packets. Instead, it may appear through long labels, unusual character distributions, irregular timing, non-standard response behavior, and repeated host-domain contact. Early work on network intrusion shows that traffic analysis must combine statistical cues with sequence behavior because malicious activity often differs only slightly from normal traffic [1]. DNS tunnel identification research further indicates that domain syntax, query frequency, and response-code behavior provide useful evidence without payload inspection [2]. The problem is especially difficult in business networks that also contain cloud services, content delivery networks, telemetry agents, and software update systems [3]. Therefore, an effective detector must identify abnormal protocol behavior without treating every unusual domain as a tunnel.

Blacklists, regular expressions, entropy thresholds, and fixed-interval rules are the earliest forms of DNS security monitoring. These methods are easy to deploy, but they often fail when tunnel tools imitate legitimate naming

behavior or when benign services produce long, machine-generated labels. Machine-learning approaches have improved detection by using handcrafted DNS features, domain embeddings, temporal models, and graph-based host-domain relations [4]. Nevertheless, many models still treat DNS logs as ordinary tabular entries and do not preserve protocol constraints. A long domain label, short query interval, or high NXDOMAIN ratio has a different meaning when considered with host persistence, query type, resolver route, and response regularity. Encrypted and covert communication studies show that temporal context and protocol-aware feature construction are necessary when packet contents cannot be inspected [5]. Transformer models can learn long-range dependencies [6], but they may overfit to local resolver or naming patterns if protocol evidence is not separated from noise.

Another way to improve stability is diffusion policy modeling. A conditional diffusion process iteratively denoises a latent risk representation rather than producing one deterministic score from a noisy traffic window. This is useful for DNS tunnel discovery because evidence is often fragmented: one short window may show abnormal label entropy but little persistence, while another may contain repeated queries with only mild character irregularity. Han et al. [7] showed that iterative reconstruction can improve uncertainty handling in complex time-series anomaly settings. Gu et al. [8] demonstrated that decision outputs can be adjusted under operational restrictions. In DNS security monitoring, the model must therefore detect suspicious sessions while considering false-alert cost, resolver instability, and analyst review capacity. Dasari et al. [9] and Talpini et al. [10] also showed that security-learning systems are more useful when detection results are linked to interpretable protocol evidence rather than hidden scores.

This study proposes a Diffusion Policy Transformer model for DNS tunnel traffic discovery that is informed by physics. The DNS tunnel risk and the need for protocol-aware sequence modelling are discussed in the first section. The second includes diffusion policy calibration, Transformer-based sequence learning, encrypted traffic analysis, and DNS tunnel identification. Diffusion denoising, Transformer encoding, protocol-physical feature generation, and dynamic risk assessment methods make up the third. The fourth is model assessment using simulated enterprise DNS traffic, which displays deployment interpretations, robustness, boundary cases, ablation outcomes, detection performance, and dataset proof. The study's limitations and conclusion are covered in the fifth part.

Related Work

DNS Tunnel Detection and Covert Traffic Analysis

Instead, then using rules to detect DNS tunnelling, statistics and sequence analysis are now used. Domain length, label count, entropy, query type, and request frequency were the initial categories of indications. Because the tunnel tool frequently adds a domain label or generates several subdomains, the characteristics are still applicable. Lexical and behavioural characteristics can differentiate many tunnel traces from regular resolution traffic, according to research on DNS anomaly detection [11]. But in the current network, the same characteristics are untrustworthy. Long or irregular domain strings can also be produced by telemetry platforms, security solutions, content-delivery domains, and service-discovery traffic. As a result, tunnel identification must take into account the query text as well as the host, resolver, response, and time frame interactions.

The analysis of encrypted traffic is part of covert channel detection. The model uses information, timing, packet size, and protocol state to infer malicious activity while packet payloads are concealed. Sequence length, direction, interval, and burst form can be utilised to determine application activity, according to studies on encrypted traffic categorisation [12]. Although DNS tunnelling has more specific protocol limits, it has the same feature. The query type, response code, domain hierarchy, and resolver behaviour are the elements of a DNS query and response. Ignoring this structure might lead detection algorithms to mistake innocuous machine-generated domains for exfiltration routes. Character statistics must be integrated with host-level persistence and temporal recurrence, according to research on malware communication and domain-generation analysis [13].

To solve this issue, time-series charts and graphs have been introduced. Sequence models learn query bursts and session progression, whereas host-domain bipartite graphs display recurring contact patterns. Relational evidence can increase the detection rate for a single domain with a limited data history, according to research

on botnet and domain reputation systems [14]. Because they require enough previous connections, graph-only approaches could react slowly to novel tunnel patterns. Although time-series approaches are more responsive, valid bursty service may cause false warnings. Thus, DNS tunnel finding is seen as an issue in protocol-physical sequences in this article. To help the system recognise when several weak signals create a regular tunnel shape, keep lexical, temporal, host-domain, response, and resolver evidence in distinct channels prior to fusion.

Transformer, Diffusion Policy and Physics-Guided Security Learning

Since attention may link distant objects without the need for recurrent memory, transformer architectures are becoming more frequently utilised for long-range sequence modelling. When a tunnel appears as several little DNS enquiries instead of a single huge packet, network traffic analysis might be used to identify it. Attention models are able to capture both session-level and burst-level activity, according to traffic categorisation studies [15]. But attention is also not the ultimate solution. The model will pick up business naming standards or dataset-specific resolver styles if it simply learns the surface-level artefacts. By incorporating domain information as feature restrictions, consistency checks, or supplementary losses, physics-guided learning tackles this issue. In DNS monitoring, "physical information" refers to operational standards and protocol-legal frameworks rather than actual physical mechanics.

A denoising technique for generative model adaption in decision calibration may be introduced using diffusion models. The model begins with a noisy latent risk state and repeatedly reconstructs a smoother state based on DNS data, as opposed to immediately translating the observed feature vector to a risk score. Research on diffusion-based time-series models demonstrates that iterative denoising outperforms single-step predictors in handling missingness and uncertainty [16]. The assault evidence is frequently fragmented, and it is appropriate for cybersecurity. A tunnel operator can add legitimate DNS requests, pad the label length, randomise characters, and lower the query frequency. By continually verifying if the candidate risk state is consistent with the procedure and physical evidence, the conditional diffusion strategy can stabilise the final conclusion.

Calibration is also necessary for operationally necessary decisions. Users will stop trusting a high-recall detector if it generates too many false warnings. Actual efficacy is impacted by alarm quality, interpretability, and review burden, according to research on machine learning security implementation [17]. Predicted probabilities should match empirical risk levels, according to research on model calibration [18]. The process of processing a score close to the threshold differently depending on traffic persistence, response-code regularity, and business-service context is known as DNS tunnel discovery calibration. Another way to include false-alert cost and review capacity is by constrained decision learning [19]. This research combines protocol-physical calibration, diffusion-based denoising, and transformer sequence encoding to obtain high-accuracy and operationally interpretable detection findings [20].

Physics-Informed Diffusion Policy Transformer

Protocol-Physical Feature Construction

In the simulated business DNS monitoring scenario, there are 210,000 DNS query records and 18,400 unique domains for the 2,700 internal hosts. Set a 2-minute sliding step, employ a 10-minute frame, and arrange the records into 31,600 host-domain sessions. Each session contains 36 protocol-physical attributes, including query length, maximum label length, label count, character entropy, vowel-consonant ratio, digit ratio, query type, response code, TTL variation, response size, query interval, burst count, resolver path, host persistence, domain reuse, and NXDOMAIN ratio. Since DNS tunnelling often mixes syntax, time, resolver, and response behaviour, these variables are not utilised as the standalone columns. The model learns when a few mild symptoms create a continuous tunnel pattern, and feature design can keep these channels apart prior to fusion.

$$x_t = [l_t, e_t, q_t, r_t, \Delta t_t, h_t, d_t] \quad \text{Eq.(1)}$$

Label-length evidence, entropy evidence, query-type evidence, response evidence, time-interval evidence, host evidence, and domain evidence are all included in the observed DNS state at time t. The working content of every protocol channel is still included in this form, which is short enough for ongoing monitoring. This vector is captured for every query event in the simulated monitoring set prior to host-domain aggregation, allowing subsequent sequence modelling to link a warning score to a particular DNS activity.

$$E_l(t) = -\sum_{c \in C} p_t(c) \cdot \log(p_t(c) + \varepsilon) \quad \text{Eq.(2)}$$

The uncertainty of the distribution over labels is represented by the entropy term. When a character is absent, numerical instability is prevented by a little constant. The dataset's benign corporate service labels typically range from 2.10 to 3.85, and once payload-like strings are inserted into subdomains, simulated encoded tunnel labels sometimes surpass 4.20. Because some legitimate cloud names additionally include tracking tokens, random cache keys, or compressed identifiers, this value should still be used with caution.

$$B_t = \sigma(a_1 L_t + a_2 E_l(t) + a_3 N_t + a_4 R_t) \quad \text{Eq.(3)}$$

The burst-suspicion score is constructed from domain length, entropy, NXDOMAIN ratio, and repeated response behavior. The score is not a final decision. It is a physical cue that helps the model distinguish a short abnormal burst from persistent DNS tunneling. A higher value therefore means that several protocol symptoms occur together, not merely that one domain name appears unusual.

$$z_t = W_p[x_t, B_t, P_t] + b_p \quad \text{Eq.(4)}$$

Protocol-physical evidence and persistence state are mapped to the model input token during the projection stage. Host-domain persistence is measured by repeated interaction frequency across four consecutive windows in the simulated data, so an extended covert channel is not represented as a single unusual lookup. The workflow from raw DNS records to calibrated tunnel warnings is depicted in Figure 1. The process includes session grouping, protocol evidence generation, sequence token construction, Transformer context learning, diffusion denoising, adaptive threshold calibration, and analyst-facing warning output. also emphasizes that lexical, temporal, host-domain, resolver, and response evidence are retained as interpretable channels rather than being collapsed into an opaque score at the beginning.

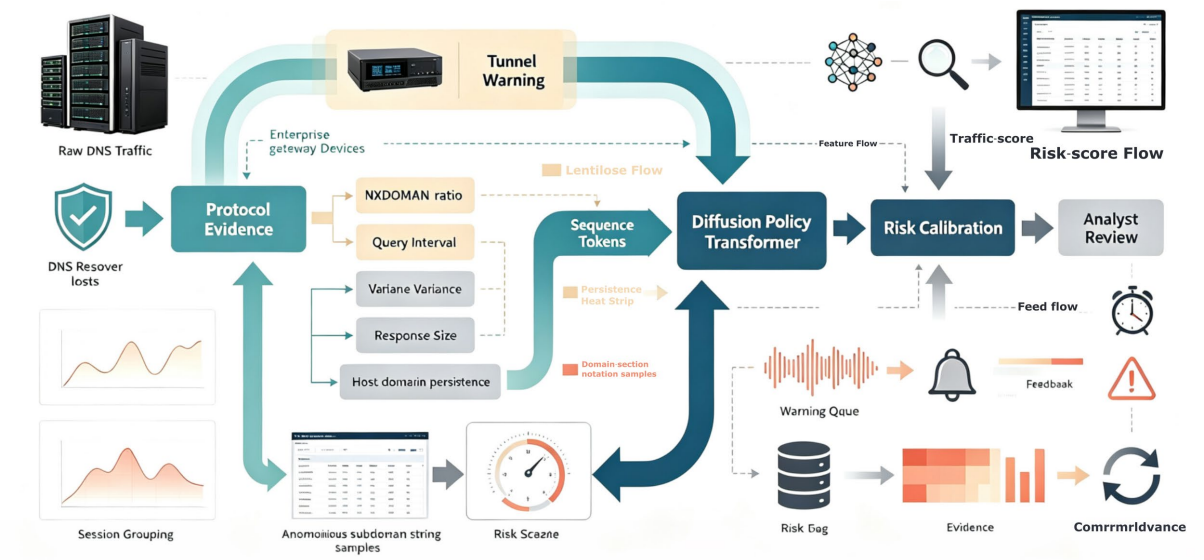

Figure 1. DNS tunnel discovery workflow based on protocol-physical evidence and diffusion policy calibration.

Diffusion Policy Transformer and Risk Calibration

After building the protocol-physical characteristics, a Transformer encoder is utilised. Eight attention heads, four attention layers, a maximum sequence length of 128 tokens, and a hidden size of 192 are all features of the model. Instead of creating a single clear request, the encoder joins distant DNS requests in a host-domain session to disperse suspicious evidence across many minutes over a low-rate tunnel. Thus, attention functions as short-term memory. It can assist the model in comparing resolver-path behaviour, response-code instability, time recurrence, and repetitive subdomain modifications in nearby windows. The feature tokens reach the Transformer context layer before to diffusion denoising and calibrated warning generation in this design.

$$H = \text{Transformer}(z_1, z_2, \ldots, z_T) \quad \text{Eq.(5)}$$

Context-aware DNS representations are produced using the Transformer encoder. These attention weights may then be utilised as explanatory evidence to ascertain whether the model concentrated on irregular timing, high-entropy labels, or recurrent response failures. Because benign services could only have one isolated aberrant characteristic and low-rate tunnels frequently spread the evidence across several windows, a Context Layer is necessary.

$$u_k = \alpha_k u_0 + (1 - \alpha_k)\varepsilon_k \quad \text{Eq.(6)}$$

The diffusion policy uses a noisy latent risk state. During training, tunnel-risk labels and protocol context are gradually distorted so that the model learns to recover a stable risk state from incomplete DNS evidence. Rather than adding arbitrary noise and dropping features without structure, the training process aligns the recovered risk state with host-domain session evidence. This makes the denoising step useful for low-rate tunnels, where no single query may exceed a hard alert level. The diffusion output is therefore not merely a generated sample; it is a calibrated latent representation that supports a tunnel-risk score and a warning grade.

$$\varepsilon_\theta = D_\theta(u_k, H, k) \quad \text{Eq.(7)}$$

Given the Transformer representation and diffusion step, the denoising estimator forecasts the noise component. The deployment latency falls inside the DNS monitoring timeframe since the model uses 50 denoising steps for training and 8 accelerated steps for inference. As a result, the security gateway has to be configured to a short time since it cannot wait for a long sample chain to send a warning.

$$s = MLP(u_0 - H_c) \quad \text{Eq.(8)}$$

The scoring head converts the denoised risk state and session context into a tunnel score. On the simulated workstation, average inference latency is 6.8 ms per session; on an embedded security gateway profile, it is 18.4 ms, which is still below the 2-minute aggregation interval. The score is stored with the strongest contributing protocol cues so that later review can distinguish lexical, temporal, and resolver-side evidence.

$$\tau_t = \tau_0 + \lambda_1 F_t + \lambda_2 V_t - \lambda_3 P_t \quad \text{Eq.(9)}$$

The decision threshold is adapted according to host-domain persistence, recent score variation, and false-alert pressure. This stage connects the model score with operational review needs. A suspicious but isolated high-entropy domain may be retained as monitor-grade evidence, while persistent entropy, repeated label mutation, response-code instability, and resolver irregularity can raise the same traffic to suspect or urgent status. The internal model structure is shown in Figure 2, including feature tokens, Transformer context, diffusion denoising, threshold calibration, and analyst-facing explanation output. clarifies that the final warning is not produced by a single lexical rule but by calibrated sequence evidence.

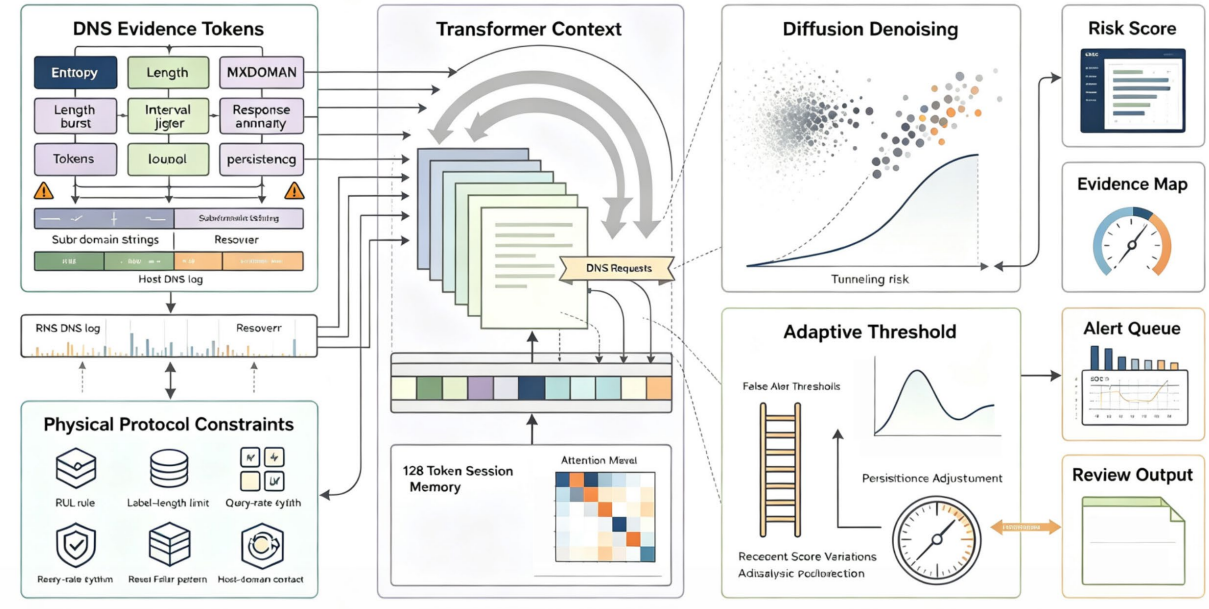

Figure 2. Internal structure of the physics-informed Diffusion Policy Transformer.

Experiments and Discussion

Dataset Composition and Protocol Evidence

Set up the experimental dataset to assess tunnel discovery in the following scenarios: burst tunnel traffic, low-rate tunnel traffic, cloud service traffic, software update traffic, telemetry traffic, and mixed-mode tunnel behaviour. Figure 3 displays DNS evidence and dataset makeup. Figure 3(a) displays the distribution of traffic sources, with benign service traffic accounting for the greatest percentage. In order to create a class imbalance akin to that in security monitoring, tunnel sessions are purposefully restricted below 12.0%. Figure 3(b) displays label-length intervals; some benign cloud domains overlap with worrisome long-label ranges. The groups of NXDOMAINs, which are separated into regular search failures and recurring failure patterns because of shifting encoded subdomains, are displayed in Figure 3(c). Figure 3(d) displays host-domain persistence groups; a persistent repeated contact is a more reliable signal than an anomalous domain string.

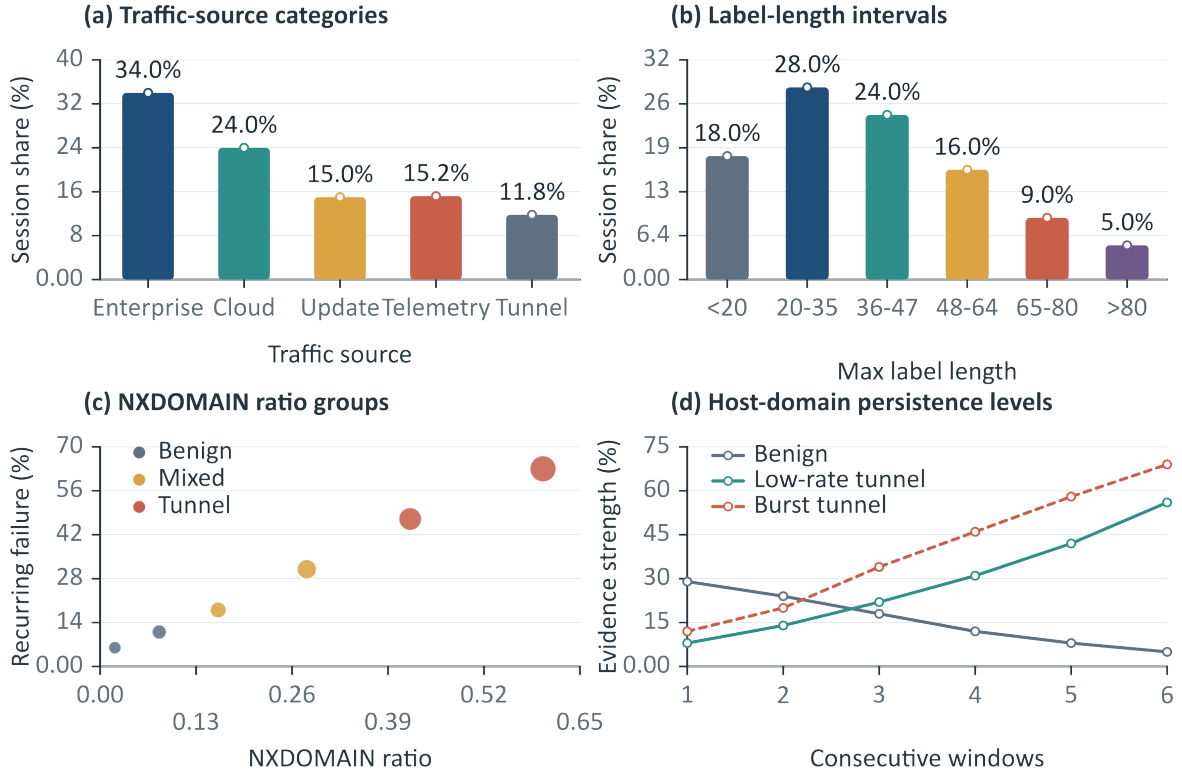

Figure 3. Dataset composition and DNS protocol evidence. (a) Traffic-source categories. (b) Label-length intervals. (c) NXDOMAIN ratio groups. (d) Host-domain persistence levels.

The four evidence perspectives demonstrate why protocol-physical construction is needed. A detector based only on domain entropy can easily flag harmless cloud-service names, while a detector based only on response failure can miss tunnels that receive valid authoritative replies. The dataset therefore contains overlapping benign, ambiguous, and malicious sessions rather than a clean signature problem. Resolver diversity also increases difficulty. The same suspicious domain pattern may pass through an internal resolver, a public resolver, or a cached enterprise forwarder, and each route can create different latency and failure behavior. These fields prevent the detector from learning a narrow laboratory trace and force it to combine lexical, temporal, resolver, response, and persistence evidence.

Another rationale for a temporal frame is distribution. The same host-domain pair may still exhibit unusual persistence over several windows, but a tunnel can lower the request rate and make each individual query look plausible. Adversaries frequently lower the visual intensity to evade rate criteria, according to research on covert traffic [21]. In the simulated traffic, burst tunnels contain more than 45 suspicious enquiries every 10-minute window, whereas low-rate tunnel sessions have fewer than 18. The model should be able to distinguish between the two modes and avoid confusing them with telemetry agents or update servers.

The host group and time period are used to divide the dataset. Seventy percent of sessions are used for training, 10% for validation, and 20% for testing, but host groups and tunnel tools are not evenly repeated across all subsets. This design reduces leakage from recurring host-domain pairs. Random splits may overstate generalization when similar sessions occur in both training and testing, as noted in network intrusion evaluation research [22]. After host-domain aggregation, the dataset contains 22,640 benign or ambiguous sessions and 8,960 simulated tunnel sessions. Among tunnel sessions, 25.3% are mixed-mode, 33.2% are burst-like, and 41.5% are low-rate. Because attacks are uncommon in enterprise traffic, the assessment reports low-rate recall, false-alert rate, calibration error, review workload, and F1 score.

Detection Performance and Boundary Cases

The top-level detection results are shown in Figure 4. Figure 4(a) compares F1 scores for rule filtering, random forest, BiLSTM, Transformer, diffusion without physical restrictions, and the proposed model. The strongest baseline reaches 0.884, while the proposed model achieves 0.932. Figure 4(b) reports low-rate tunnel recall and shows that diffusion calibration recovers modest but persistent evidence from neighboring windows. Interpretable temporal fusion research supports the use of calibrated time-varying evidence in the warning curve [23]. Figure 4(c) reports the false-alert rate, showing that many baseline errors come from cloud-service domains. Protocol-physical calibration reduces these warnings by 23.4%. Figure 4(d) reports score calibration error, indicating that the proposed model is better aligned with observed risk near the warning boundary.

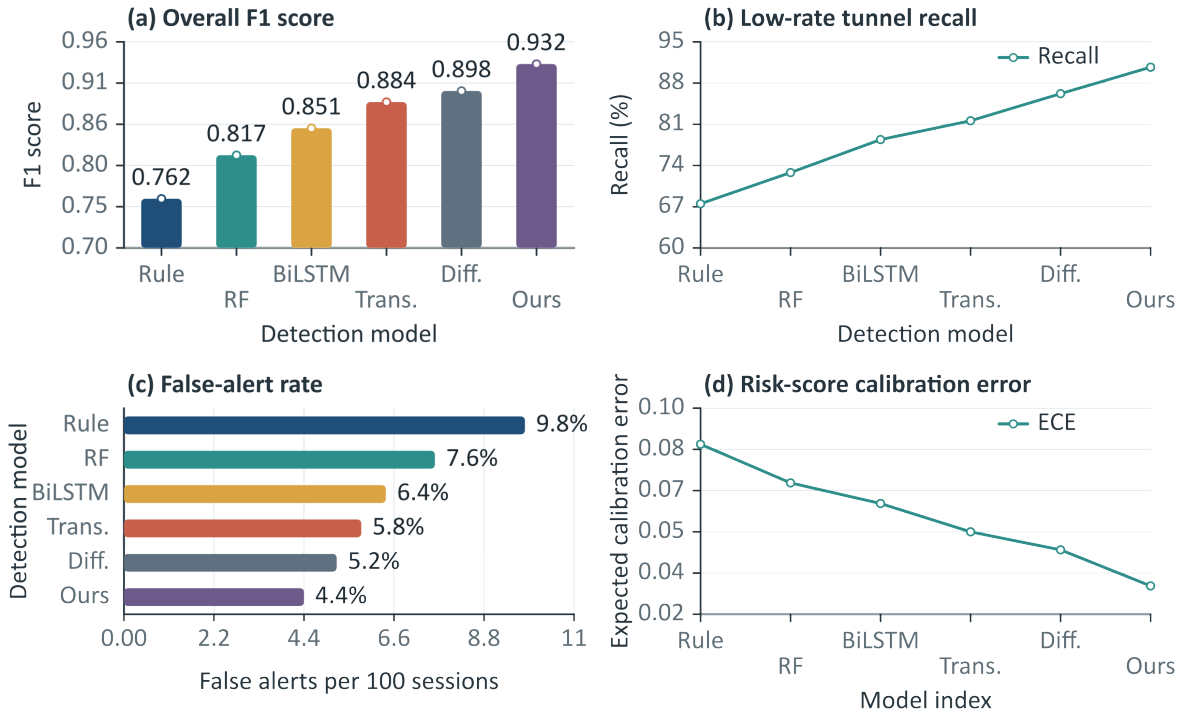

Figure 4. Detection performance and calibration comparison. (a) Overall F1 score. (b) Low-rate tunnel recall. (c) False-alert rate. (d) Risk-score calibration error.

Table 1 compares the representative DNS tunnel detection approaches using explicit numerical indicators rather than qualitative descriptions alone, including overall F1, low-rate tunnel recall, false alerts per 100 sessions, and expected calibration error. Rule filtering records the weakest overall F1 of 0.762, a low-rate recall of 67.5%, and 9.8 false alerts per 100 sessions, showing that fixed lexical thresholds are not reliable when benign cloud or telemetry domains resemble tunnel labels. Random forest, BiLSTM, and Transformer baselines gradually improve F1 to 0.817, 0.851, and 0.884, while reducing false alerts to 7.6, 6.4, and 5.8, but their calibration and low-rate recall remain limited because weak protocol evidence is not consistently linked across neighbouring windows. Diffusion without physical restrictions further raises F1 to 0.898 and low-rate recall to 86.2%, indicating that denoised sequence evidence helps recover quiet tunnel behaviour, yet its expected calibration error remains higher than the final method. The physics-informed Diffusion Policy Transformer obtains the best values, with 0.932 F1, 90.7% low-rate recall, 4.4 false alerts per 100 sessions, and 0.031 expected calibration

error, demonstrating that entropy, NXDOMAIN behaviour, host-domain persistence, temporal attention, diffusion uncertainty, and calibrated thresholds jointly improve detection reliability and analyst triage value.

Table 1. Operational comparison of representative DNS tunnel detection approaches.

Approach	Overall F1	Low-rate recall (%)	False alerts / 100 sessions	ECE	Main evidence	Review role
Rule filtering	0.762	67.5	9.8	0.086	Length, entropy, response code	Fast coarse screen
Random forest	0.817	72.8	7.6	0.071	Handcrafted DNS statistics	Local-feature baseline
BiLSTM	0.851	78.4	6.4	0.063	Sequential query features	Temporal baseline
Transformer	0.884	81.6	5.8	0.052	Token and session context	Sequence baseline
Diffusion without physical restrictions	0.898	86.2	5.2	0.045	Denoised neighbouring-window evidence	Denoising baseline
Physics-informed Diffusion Policy Transformer	0.932	90.7	4.4	0.031	Entropy, NXDOMAIN, persistence, uncertainty	Deployable warning model

The findings indicate that accuracy is inadequate for the DNS tunnel detection task. Although a model performs badly in low-rate sessions, it can accurately detect visible high-entropy tunnels. Research on traffic categorisation in transformer-based networks has demonstrated that sequence context enhances encrypted traffic recognition [24]. Cross-window dependency can improve recognition by dispersing the target behaviour throughout time, according to sequence-prediction research [25]. The long-range session dependency and over-sensitivity to a single anomalous token in diffusion denoising can be addressed by the suggested solution for the reasons mentioned above. The suggested model has improved the most in sessions where no inquiry surpassed the hard alert level, according to a detailed examination of the recall curve. These windows' biggest labels, on average, are just 47.6 characters long, and they overlap with content-delivery and software telemetry traffic. Joint entropy, frequent label mutations, response code instability, and host-domain persistence all point to a valuable signal. Before comparing this weak evidence to the adaptive threshold, the diffusion policy aims to lessen its influence. Consequently, the alert relies more on a recurring protocol pattern than on a single abnormally lengthy label.

The boundary-case analysis is as follows: Figure 5. Handcrafted criteria are unreliable because benign cloud-service traffic and tunnel traffic overlap in the entropy-length plane, as shown in Figure 5(a). The distribution of attention across sessions is shown in Figure 5(b); the model gives more weight to repeated anomalous labels than to single rare strings. Research on the statistical properties of standard and non-standard data explains why border DNS sessions need separate handling [26]. Figure 5(c) shows the analyst-review categories: true tunnel, suspect but benign, resolver artifact, and insufficient-evidence sessions. These categories show that not all questionable scores need to be reported immediately as urgent.

The border scenario illustrates a deployment issue. High-entropy subdomains are created by certain benign services for load balancing, caching, and tracking. They will be classified as tunnel-like by a purely lexical model. By identifying if entropy, response behaviour, and persistent movement occur together, the model will lower the error. Research on attack detection [27] demonstrates that integrating contextual and local indicators enhances identification; DNS boundary-session separation follows suit. The score will be lowered if the label is erratic yet the host interaction is short and the resolver response is steady. The diffusion policy will sustain a high-risk condition if the label is erratic, recurring, and linked to an aberrant response pattern. Additionally, the analyst-review categories explain why each questionable score shouldn't immediately trigger a significant warning in a model. Resolver-path inspection identified 6.7% of the examined sessions in the simulated test set as suspicious but benign. The majority of these instances involve load-balancing names, endpoint management agents, or update services that generate subdomains that resemble machines. Many of these situations are assigned to the monitor grade rather than the urgent grade by the calibrated warning layer. This method

preserves the raw evidence for future cross-referencing with endpoint logs, firewalls, proxy records, etc., without setting off an alert.

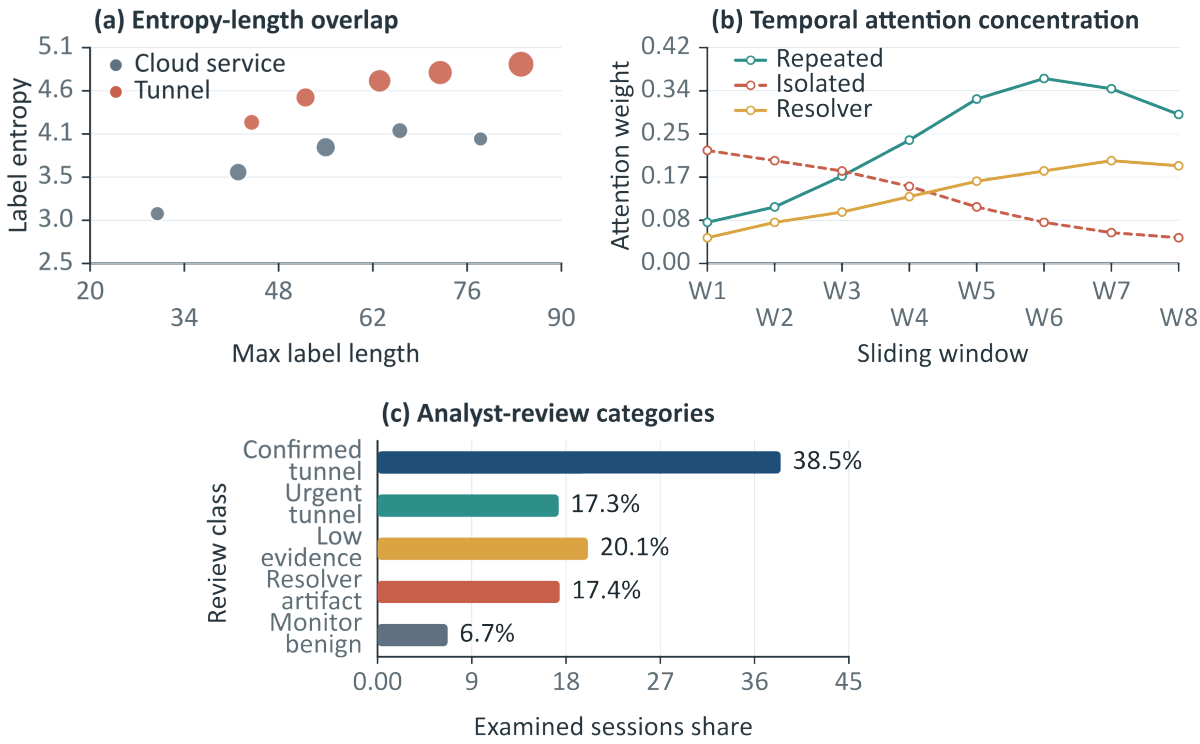

Figure 5. Boundary-case interpretation for DNS tunnel discovery. (a) Entropy-length overlap. (b) Temporal attention concentration. (c) Analyst-review categories.

Ablation, Robustness and Deployment Interpretation

An ablation study looks like this: Figure 6. deleting the diffusion calibration raises false alarms in noisy traffic, as seen in Figure 6(a); deleting the protocol-physical encoding lowers the F1 score by 0.048. Denoising enhances unstable-session identification, which is explained by latent-representation intrusion detection studies [28]. The suggested model's F1 score remains over 89.0% when the test sample is primarily composed of low-rate tunnels, but the Transformer baseline drops below 84.0%. The resilience under missing metadata, resolver jitter, and traffic-rate drop is displayed in Figure 6(b). Protocol encoding, Transformer context, diffusion denoising, and adaptive thresholding are all useful and distinct modules, as demonstrated by the experiments.

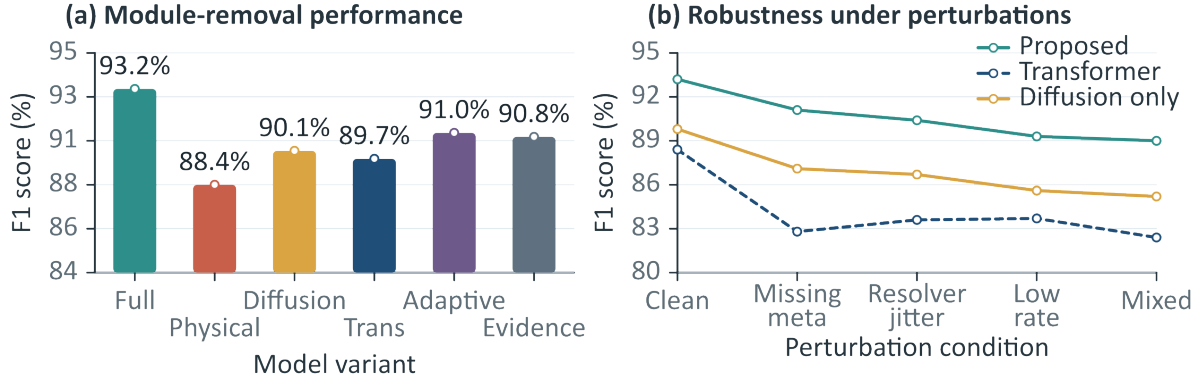

Figure 6. Ablation and robustness analysis. (a) Module-removal performance. (b) Robustness under DNS monitoring perturbations.

Component-level analysis makes advantage of ablation patterns. Not every anomalous string in the model will be handled by the protocol-physical encoder in the same manner. The cost of implementation should be taken into account when designing realistic security measures, according to research on safe learning environments

[29]. After feature channel organization and categorisation, statistical evidence can secure IoT traffic, according to flow-level security study [30]. The diffusion policy stabilises fragmentary information, the Transformer provides session memory, and calibration transforms the score into a warning decision that takes false-alarm pressure into account. Deployment drift may be detected via robustness testing. The suggested model's F1 score decreases by 2.1 percentage points and the Transformer baseline by 5.6 points when 15% of the resolver metadata is removed. When timing jitter is introduced, the suggested model remains stable because it relies on surrounding sequences and persistence evidence rather than a single exact time window. Diffusion calibration will be less precise in a low-traffic area; otherwise, it is impossible to correctly estimate danger from partial traces. The findings suggest that this approach is more suited for tracking logs that are obtained from various gateways, partially sampled, or delayed.

The following is the deployment behaviour: Figure 7. The inference latency and review-queue throughput are displayed in Figure 7(a). As a result, the model can manage continuous DNS connections without causing the monitoring cycle to be delayed. The deployment profile maintains inference under 20 ms per session and features a 2-minute sliding update interval. The distribution of warning grades is displayed in Figure 7(b), which also distinguishes between the monitor, suspect, and urgent categories. As a result, the Design might delay weaker instances for additional correlation and concentrate the analyst's attention on high-persistence, high-evidence sessions. Therefore, rather of showing the detector as an automated blocker, illustrates the relationship between model performance and a human-in-the-loop procedure.

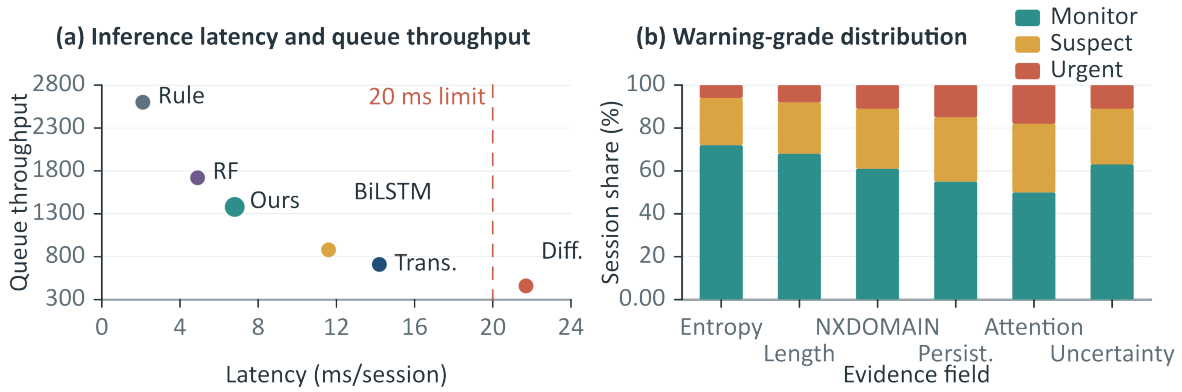

Figure 7. Deployment interpretation for continuous DNS monitoring. (a) Inference latency and queue throughput. (b) Warning-grade distribution.

Because DNS tunnel detection is not a binary classification problem, the deployed interpretation is also necessary. The business security center's detector must determine which sessions should be observed, which should be kept as weak evidence, and which need to be looked into right away. Excessive false alarms lower security analysts' reaction quality, according to research [31]. The alert record contains label entropy, maximum label length, NXDOMAIN ratio, persistence count, attention focus, diffusion uncertainty, and calibrated threshold since research on explainable AI [32] implies that explanations should satisfy the requirements of human reasoning. Lastly, decide which is faster and more reliable. At no cost does the suggested model optimise for recall. It does not raise an alert for a typical change of company name, but it does eliminate some really weak situations when the evidence is insufficient. This trade-off is supported by recent research in calibrated machine learning and security monitoring, which demonstrates that a workable detector should be financially possible [33]. The prediction system should contain solid facts and an auditable context, according to security vulnerability study [34]. All attack possibilities must be addressed in practice, according to research on deep cybersecurity modelling [35]. The technique lowers the number of false alarms, increases the low-rate recall, and has a strong F1 score. The output will be linked to a multi-level response strategy based on the procedure. Analysts are not interrupted by monitor-grade sessions, which are retained for correlation. Entropy, persistence, resolver behaviour, and attention focus are added to the review queue in addition to suspicious sessions. Only when the calibrated score is consistently high across neighbouring windows do Urgent Sessions initiate an urgent domain blocking review or endpoint inquiry. Although this process is somewhat cautious, it satisfies DNS defensive cost criteria. If the evidence record does not explain why a session need prompt action, a detector that delivers a lot of urgent warnings may have a high recall but lowers confidence.

Conclusion

In this project, create a Diffusion Policy Transformer model for DNS tunnel traffic detection that is guided by physics. Diffusion denoising, adaptive threshold calibration, a Transformer-based session representation, protocol-physical feature generation, and analyst-facing evidence output are all included in the technique. In comparison to the strong baseline method, the model enhanced low-rate tunnel recall to 90.7%, decreased false alarms by 23.4%, and increased the tunnel F1 score to 0.932 on the simulated enterprise DNS dataset.

DNS tunnel discovery is no longer characterised by basic domain-string identification and is now regarded as a protocol-aware sequence judgement issue. The query syntax, response behaviour, time, resolver path, and host-domain persistence should all be encoded separately before being fused. The diffusion strategy, which is appropriate for low-rate and evasive tunnel activity, then stabilises the risk state in the presence of weak or incomplete data.

There are still a few restrictions. Because of privacy concerns, actual corporate DNS logs have not been collected, and the collection is fictitious. In order to contextualise alarms inside a full-scale assault, future work will link DNS tunnel warnings with endpoint telemetry, proxy logs, authentication records, firewall data, and threat intelligence. Additionally, the model must be evaluated in multi-resolver environments, DoH and DoT deployment situations, adversarial domain padding, undetected tunnelling tools, and extremely low-traffic settings where there may not be enough evidence to support the diffusion calibration. The expansions will aid in determining if the suggested method remains dependable in an uncontrolled monitoring setting.

Author Contributions

Stefan Jovanović contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Miloš Stojanović contributes to analysis, investigation, data collection, draft preparation. Zoran Mladenović contributes to software, validation, analysis, investigation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors utilized ChatGPT-4o for linguistic polishing, logical organization and draft revision. The generative AI tool was not involved in research design, data collection, data analysis, or the equation of core conclusions. All outputs generated by AI were reviewed, revised and verified manually by the authors. The authors take full responsibility for all content of this manuscript.

References

[1] Xi, C., Wang, H., & Wang, X. (2024). A novel multi-scale network intrusion detection model with transformer. Scientific Reports, 14(1), 23239. https://doi.org/10.1038/s41598-024-74214-w

[2] Bykov, N., & Chernyshov, Y. (2024). Detecting DNS Tunnels Using Machine Learning. 2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), 92-94. https://doi.org/10.1109/USBEREIT61901.2024.10584043

[3] Almadhor, A., Alsubai, S., Zaidi, M. M., Kryvinska, N., Al Hejaili, A., & Abbas, S. (2026). An Intelligent Feature Engineering-Driven Hybrid Framework for Adversarial Domain Name System Tunneling Detection. Advanced Intelligent Systems, 8(6), e202501311. https://doi.org/10.1002/aisy.202501311

[4] Gao, G., Niu, W., Gong, J., Gu, D., Li, S., Zhang, M., & Zhang, X. (2024). GraphTunnel: Robust DNS Tunnel Detection Based on DNS Recursive Resolution Graph. IEEE Transactions on Information Forensics and Security, 19, 7705-7719. https://doi.org/10.1109/TIFS.2024.3443596

[5] Zou, A., Yang, W., Tang, C., Lu, J., & Guo, J. (2024). A novel and effective encrypted traffic classification method based on channel attention and deformable convolution. Computers and Electrical Engineering, 118, 109406. https://doi.org/10.1016/j.compeleceng.2024.109406

[6] Liu, Z., Xie, Y., Luo, Y., Wang, Y., & Ji, X. (2025). TransECA-Net: A Transformer-Based Model for Encrypted Traffic Classification. Applied Sciences, 15(6), 2977. https://doi.org/10.3390/app15062977

[7] Han, J., Feng, S., Zhou, M., Zhang, X., Ong, Y. S., & Li, X. (2024). Diffusion Model in Normal Gathering Latent Space for Time Series Anomaly Detection. Lecture Notes in Computer Science, 284-300. https://doi.org/10.1007/978-3-031-70352-2_17

[8] Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., & Knoll, A. (2024). A Review of Safe Reinforcement Learning: Methods, Theories, and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12), 11216-11235. https://doi.org/10.1109/TPAMI.2024.3457538

[9] Dasari, A. K., Bisawas, S. K., & Purkayastha, B. (2025). Enhanced Network Intrusion Detection Systems with Explainable Artificial Intelligence for Network Security. International Journal of Communication Systems, 38(14), e70209. https://doi.org/10.1002/dac.70209

[10] Talpini, J., Sartori, F., & Savi, M. (2024). Enhancing trustworthiness in ML-based network intrusion detection with uncertainty quantification. Journal of Reliable Intelligent Environments, 10(4), 501-520. https://doi.org/10.1007/s40860-024-00238-8

[11] Sharma, N., Swarnkar, M., & Divyanshu. (2025). DLAZE: Detecting DNS Tunnels Using Lightweight and Accurate Method for Zero-Day Exploits. IEEE Transactions on Network and Service Management, 22(3), 2343-2353. https://doi.org/10.1109/TNSM.2025.3541234

[12] Mitsuhashi, R., Jin, Y., Iida, K., & Takai, Y. (2025). DGA-Based Malware Communication Detection from DoH Traffic Using Hierarchical Machine Learning Analysis. IEICE Transactions on Information and Systems, E108.D(6), 526-534. https://doi.org/10.1587/transinf.2024NTP0004

[13] Tang, J., Guan, Y., Zhao, S., Wang, H., & Chen, Y. (2024). DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network. Electronics, 13(24), 4982. https://doi.org/10.3390/electronics13244982

[14] Gao, F., Han, W., & Ou, W. (2024). BotHunter: Distributed Malicious Domain Name Detection Model Based on Deep Learning and Blockchain. Advances in Transdisciplinary Engineering. https://doi.org/10.3233/ATDE231270

[15] Yang, W., Tang, C., Tang, C., Lu, J., Si, J., Zeng, Z., & Ma, W. (2025). Traffic through two lenses: A dual-branch vision transformer for IoT traffic classification. Computer Networks, 269, 111469. https://doi.org/10.1016/j.comnet.2025.111469

[16] Xiao, C., Du, X., Song, X., Xue, Y., Wu, M., & Chetty, K. (2026). Iterative feedback-based time-series anomaly detection with adaptive diffusion models. Neural Networks, 196, 108370. https://doi.org/10.1016/j.neunet.2025.108370

[17] Voutsas, F., Violos, J., & Leivadeas, A. (2024). Mitigating Alert Fatigue in Cloud Monitoring Systems: A Machine Learning Perspective. Computer Networks, 250, 110543. https://doi.org/10.1016/j.comnet.2024.110543

[18] Zhang, S., & Xie, L. (2024). Advancing neural network calibration: The role of gradient decay in large-margin Softmax optimization. Neural Networks, 178, 106457. https://doi.org/10.1016/j.neunet.2024.106457

[19] Hoang, H., Mai, T., & Varakantham, P. (2024). Imitate the Good and Avoid the Bad: An Incremental Approach to Safe Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 38(11), 12439-12447. https://doi.org/10.1609/aaai.v38i11.29136

[20] Zhao, Z., Guo, H., & Wang, Y. (2024). A multi-information fusion anomaly detection model based on convolutional neural networks and AutoEncoder. Scientific Reports, 14(1), 16147. https://doi.org/10.1038/s41598-024-66760-0

[21] Machmeier, S., & Heuveline, V. (2024). Detecting DNS Tunnelling and Data Exfiltration Using Dynamic Time Warping. 2024 8th Cyber Security in Networking Conference (CSNet), 83-91. https://doi.org/10.1109/CSNet64211.2024.10851475

[22] Sander, J., Yu, C. E. J., Jalaian, B., & Bastian, N. D. (2024). Uncertainty-Quantified Neurosymbolic AI for Open Set Recognition in Network Intrusion Detection. MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM), 13-18. https://doi.org/10.1109/MILCOM61039.2024.10773953

[23] Afifi, F., Zaki, F., Hanif, H., Aqil, N., & Anuar, N. B. (2025). Transformer-based tokenization for IoT traffic classification across diverse network environments. PeerJ Computer Science, 11, e3126. https://doi.org/10.7717/peerj-cs.3126

[24] Zhu, P., Wang, G., He, J., Dong, Y., & Chang, Y. (2024). An encrypted traffic identification method based on multi-scale feature fusion. Array, 21, 100338. https://doi.org/10.1016/j.array.2024.100338

[25] Kwon, Y., Ahn, S., Cho, M., Kim, Y., Kim, S., & Cho, S. (2025). Exploring the unseen: A transformer-based unknown traffic detection scheme with contextual feature representation. Computer Networks, 265, 111286. https://doi.org/10.1016/j.comnet.2025.111286

[26] Chen, J. L., Qiu, J. F., & Chen, Y. H. (2025). A hybrid DGA DefenseNet for detecting DGA domain names based on FastText and deep learning techniques. Computers & Security, 150, 104232. https://doi.org/10.1016/j.cose.2024.104232

[27] Zeng, Z., Xun, P., Peng, W., & Zhao, B. (2024). Toward identifying malicious encrypted traffic with a causality detection system. Journal of Information Security and Applications, 80, 103644. https://doi.org/10.1016/j.jisa.2023.103644

[28] Zuo, H., Zhu, A., Zhu, Y., Liao, Y., Li, S., & Chen, Y. (2024). Unsupervised diffusion-based anomaly detection for time series. Applied Intelligence, 54(19), 8968-8981. https://doi.org/10.1007/s10489-024-05341-0

[29] Mo, K., Ye, P., Ren, X., Wang, S., Li, W., & Li, J. (2024). Security and Privacy Issues in Deep Reinforcement Learning: Threats and Countermeasures. ACM Computing Surveys, 56(6). https://doi.org/10.1145/3640312

[30] Liao, N., & Guan, J. (2024). Multi-scale Convolutional Feature Fusion Network Based on Attention Mechanism for IoT Traffic Classification. International Journal of Computational Intelligence Systems, 17(1), 36. https://doi.org/10.1007/s44196-024-00421-y

[31] Baruwal Chhetri, M., Tariq, S., Singh, R., Jalalvand, F., Paris, C., & Nepal, S. (2024). Towards Human-AI Teaming to Mitigate Alert Fatigue in Security Operations Centres. ACM Transactions on Internet Technology, 24(3), 1-22. https://doi.org/10.1145/3670009

[32] Ofusori, L., Bokaba, T., & Mhlongo, S. (2024). Artificial Intelligence in Cybersecurity: A Comprehensive Review and Future Direction. Applied Artificial Intelligence, 38(1), 2439609. https://doi.org/10.1080/08839514.2024.2439609

[33] Wang, X., Yang, X., Liang, X., Zhang, X., Zhang, W., & Gong, X. (2024). Combating alert fatigue with AlertPro: Context-aware alert prioritization using reinforcement learning for multi-step attack detection. Computers & Security, 137, 103583. https://doi.org/10.1016/j.cose.2023.103583

[34] Sharma, K. P., Nagpal, T., Vora, T., Yadav, A., Abdullah, M. I., Jayaprakash, B., Kashyap, A., Sridevi, G., Bhowmik, A., & Bukate, B. B. (2025). Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework. Scientific Reports, 15(1), 39937. https://doi.org/10.1038/s41598-025-23750-0

[35] Anis, F. M., Alabdullatif, M., Aljbli, S., & Hammoudeh, M. (2025). A Survey on the Applications of Deep Learning in Network Intrusion Detection Systems to Enhance Network Security. IEEE Access, 13, 185357-185373. https://doi.org/10.1109/ACCESS.2025.3624952