Electronics and Information Technologies, 2026, 33, 57–70
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.7

EVALUATION OF THE PERFORMANCE OF A MULTI-LEVEL MODEL FOR ANOMALOUS DNS QUERY DETECTION

Andrii Senyk
Department of Information and Communication Technologies,
Lviv Polytechnic National University,
12 Stepan Bandera Str., Lviv, 79013, Ukraine

Senyk, A. (2026). Evaluation of the Performance of a Multi-Level Model for Anomalous DNS Query Detection. *Electronics and Information Technologies*, 33, 57–70. https://doi.org/10.30970/eli.33.5

ABSTRACT

Background. In modern network security systems, DNS (Domain Name System) traffic has become an increasingly attractive vector for covert data exfiltration and command-and-control communication. Existing machine learning methods frequently suffer from limited adaptability to novel attack patterns and an imbalance between detection accuracy and false positive rates.

Materials and Methods. This study proposes TunnelEye, a multi-level detection method for malicious DNS queries that integrates statistical feature analysis, structural n-gram modeling, and anomaly detection. Statistical properties of domain names, including string length, entropy, and alphanumeric ratio, are used for initial discrimination between benign and suspicious queries. Structural analysis based on character n-grams enables the identification of local patterns associated with encoded data such as Base32 and Base64. An autoencoder trained exclusively on legitimate DNS queries is employed as an independent anomaly detector to identify previously unseen and zero-day attacks. The supervised TunnelEye classifier and the autoencoder operate in parallel, each using an independently optimized F1-score based threshold to determine anomalous DNS queries.

Results and Discussion. Experimental evaluation using standard machine learning metrics (precision, recall, F1-score, ROC-AUC, PR-AUC, and false positive rate) demonstrates that TunnelEye consistently outperforms baseline statistical models and standalone autoencoders. The proposed method achieves high precision and recall while maintaining a minimal false positive rate. Experimental results show that TunnelEye achieves an average precision, recall, and F1-score of approximately 0.99, outperforming the baseline statistical model by more than 10% and significantly reducing the false positive rate.

Conclusion. TunnelEye provides a comprehensive and adaptive solution for malicious DNS query detection by combining supervised and unsupervised learning with dynamic threshold optimization. Its ability to balance detection accuracy and false positive reduction makes it well suited for deployment in modern enterprise cybersecurity systems for real-time DNS traffic monitoring.

Keywords: DNS traffic, anomaly detection, machine learning, multi-level model.

INTRODUCTION

In modern computer networks, DNS plays a crucial role in mapping domain names to IP addresses. However, this complex infrastructure is increasingly exploited by malicious actors to create covert communication channels and data tunnels. These techniques allow the transmission of harmful information directly through DNS queries,

bypassing traditional security systems and filters. Consequently, there is a growing demand for reliable algorithms to detect these threats in real time [1–4].

Traditional methods for detecting malicious DNS queries typically rely on fixed rules or statistical properties, such as domain name length or the frequency of certain characters. While these methods can detect simple forms of tunneling attacks, they are generally ineffective against sophisticated or adaptive attacks that disguise traffic as legitimate requests. This leads to some malicious queries going unnoticed, increasing the risk of network system compromise. Modern machine learning approaches, such as random forests or neural networks, enable the interpretation of more complex patterns in DNS traffic. However, even these methods face generalization challenges: models trained on historical data may be ineffective against new types of tunneled queries. This highlights the need for adaptive, multi-leveled approaches that consider both statistical and structural properties of domain names. A key challenge is the lack of high-quality training data for malicious queries. Known data collection methods are often limited or do not include the latest tunneling techniques, making it difficult to develop models capable of detecting zero-day attacks (i.e., previously unseen attacks). Therefore, the goal of this research is to create a system that can learn from reliable data and detect anomalies in new queries.

Another important issue is the balance between detection accuracy and completeness. Using fixed thresholds for classification often results in high rates of false positives or false negatives. For practical security systems, it is essential to develop an adaptive threshold adjustment mechanism that maintains an optimal balance under varying network traffic conditions. The structural properties of domain names, such as n-gram character sequences, remain underutilized in many traditional methods. These features allow the detection of local and internal patterns in malicious domain names, particularly when encoded using Base32 or Base64. Combining these features with statistical properties can significantly enhance the effectiveness of detection systems.

An additional challenge is ensuring scalability and real-time monitoring capabilities. Enterprise network systems process enormous volumes of DNS queries; therefore, algorithms must be efficient enough to operate in large-scale networks without introducing network delays. This requires an approach that combines accuracy with low computational complexity and adaptability to fluctuating loads [5–7]. Analysis of existing studies shows that most methods for detecting malicious DNS queries either rely on fixed statistical rules or use standalone machine learning models without adaptive mechanisms.

This work proposes the TunnelEye method, which integrates known components (statistical analysis, n-grams, and an autoencoder) into a novel decision-making architecture. The novelty lies in the way these components are constructed and used within a unified experimental environment. The main modifications include: integration of statistical and n-gram features into a single feature model; use of an ensemble Random Forest classifier instead of heuristic rules; inclusion of an autoencoder as an independent anomaly detector; implementation of an adaptive classification threshold selection mechanism based on the F1-score.

MATERIALS AND METHODS

Analysis of Problems and Formulation of the Research Task

This study proposes a multi-leveled adaptive method, TunnelEye, which implements a multi-leveled model for detecting malicious DNS queries based on statistical, structural, and anomalous features. This combination addresses the limitations of traditional methods, creating a more reliable and effective monitoring system.

In contrast to existing approaches to DNS tunneling detection, which typically focus either on statistical properties of domain names (such as length, entropy, or character

distribution) or on the isolated application of machine learning techniques, the proposed TunnelEye method integrates multiple levels of analysis within a unified architecture. Statistical models (e.g., rule-based approaches or Random Forest classifiers using simple numerical features) are effective at detecting primitive forms of tunneling, but they exhibit poor generalization in more sophisticated obfuscation scenarios. Methods based on n-gram analysis are capable of identifying local structural patterns (e.g., Base32 or Base64 encodings); however, in the absence of broader statistical context, they often suffer from elevated false-positive rates.

Autoencoder-based approaches commonly employed for anomaly detection in DNS traffic typically operate as standalone solutions and rely exclusively on reconstruction error. This reliance limits their accuracy in cases where malicious traffic closely mimics legitimate behavior. In contrast to these approaches, TunnelEye implements a parallel architecture in which a supervised classifier (utilizing statistical and n-gram features) and an autoencoder operate independently but are evaluated within a unified experimental framework with adaptive threshold optimization. As a result, the proposed method combines the strengths of signature-based, statistical, and anomaly-based analysis while mitigating their individual limitations.

Key Concepts in Malicious DNS Query Detection

DNS forms the foundation of the Internet by translating domain names into IP addresses. This system allows users to access websites and services using human-readable names rather than numerical addresses. However, due to its widespread use, malicious actors often exploit DNS for covert data exchange and tunneling, making DNS traffic monitoring critically important for network security.

Malicious DNS queries vary in nature, ranging from simple spoofing of legitimate domain names to sophisticated DNS tunneling, where data is encoded within a domain name and transmitted via the query. These queries are difficult to detect using traditional filtering methods because they can appear as normal, legitimate traffic, increasing the risk of hidden attacks. A key concept in detecting malicious DNS queries is the use of analytical signatures (features that can distinguish legitimate queries from malicious ones). These features include statistical properties of domain names, such as string length, entropy, the number of domain levels, and the ratio of digits to letters.

These properties allow the assessment of the randomness and complexity of domain names, which are often present in malicious queries. Additionally, processing large volumes of DNS queries in real time requires high computational efficiency and algorithmic optimization [9–12]. The general scheme for detecting malicious DNS queries is shown in **Fig. 1**.

Proposed Monitoring Method

The proposed method, named TunnelEye, is designed to detect covert communication channels in DNS traffic, specifically data tunneling through domain name queries. TunnelEye is developed as a standalone implementation for DNS tunneling detection and is used for comparison with a baseline model and an autoencoder. Unlike traditional methods, it combines several independent analysis mechanisms, enhancing the system's resilience to various obfuscation techniques. At the first level, the method employs traditional statistical features of domain names, including length, entropy, the number of domain levels, and the ratio of digits to letters. These features accurately describe the overall complexity and randomness of a string and can often distinguish legitimate domain names from those generated or used for tunneling.

The second level of the method relies on n-gram character analysis, which can detect local structural patterns in domain names. This allows the effective identification of malicious patterns related to data encodings (such as Base32 or Base64) that are typically not detectable through statistical analysis alone.

The third component of the system is an autoencoder trained specifically on “clean” samples of legitimate domain names. It acts as an anomaly detector, using reconstruction error as an indicator of deviation. This enables the detection of new types of malicious queries that were not present in the training datasets, ensuring the method’s robustness against zero-day attacks.

Furthermore, the method provides an adaptive classification threshold mechanism. Instead of using fixed values, the system uses the F1 score to determine the optimal threshold, balancing precision and recall based on specific deployment conditions. This improves the practical effectiveness of the system across a wide range of scenarios.

Thanks to this architecture, TunnelEye can serve as a multi-leveled DNS query monitoring platform and can be integrated into enterprise security systems for real-time network traffic analysis. This approach combines signature accuracy, statistical methods, and anomaly detection flexibility to provide comprehensive protection. Therefore, TunnelEye can be defined as a hybrid method for detecting covert channels in DNS. It integrates statistical analysis, n-gram-based structural analysis, and anomaly detection via an autoencoder with adaptive classification thresholds, enhancing network security capabilities and effectively detecting both known and previously unseen malicious activity.

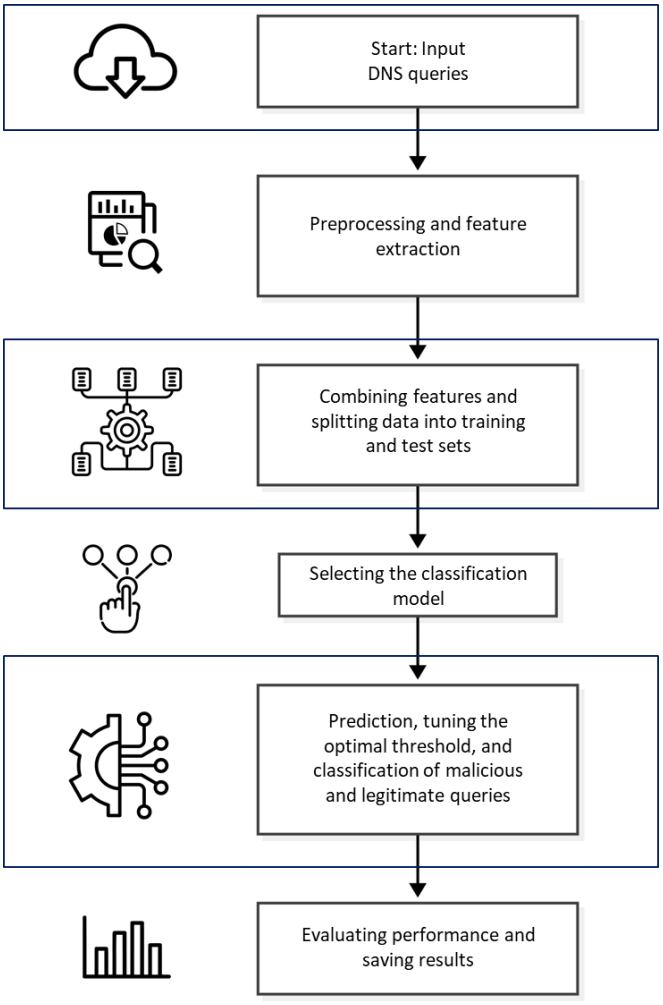

Fig. 1. General Scheme for Malicious DNS Query Detection.

Scientific Novelty of the Proposed Method

The scientific novelty of the method lies in the combination of statistical and symbolic n-gram features within a single detection framework; the parallel use of supervised and anomaly-based processing of DNS queries; and the study of the impact of the classification threshold on the effectiveness of malicious DNS query detection. The autoencoder is used as the anomaly detection mechanism, trained exclusively on legitimate DNS queries. This enables the detection of entirely new types of malicious patterns that are not present in the training samples. The block diagram of the proposed monitoring method is shown in **Fig. 2**.

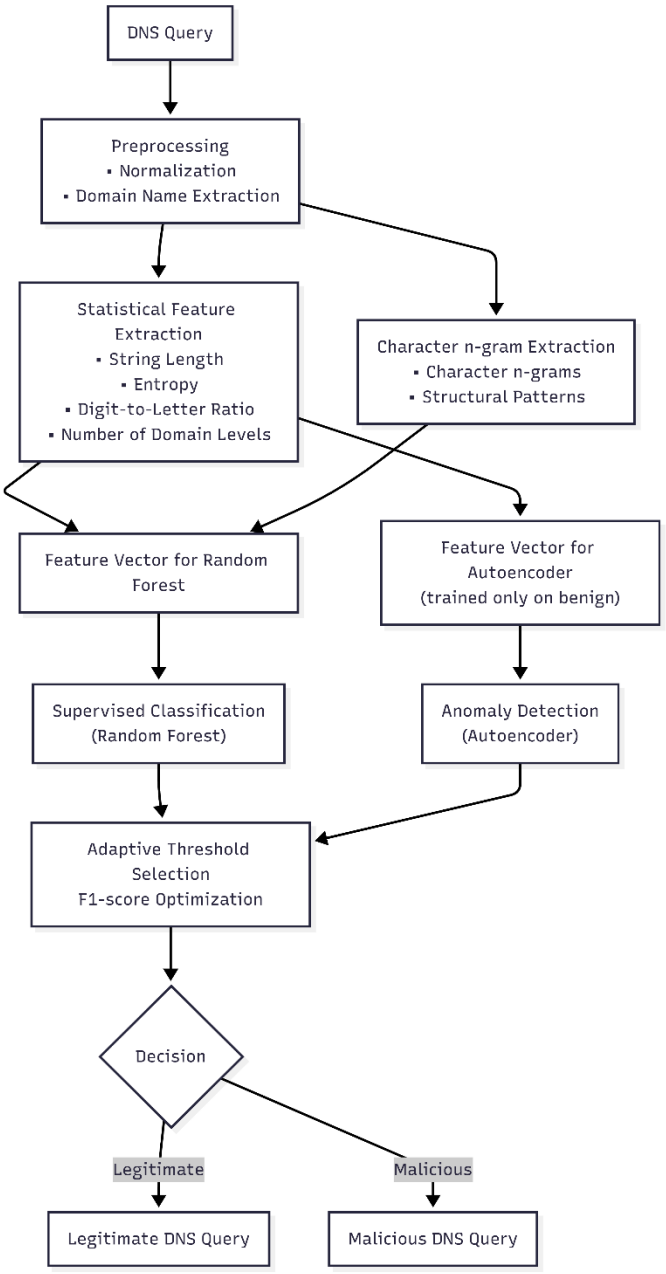

Fig. 2. Block Diagram of the Proposed Monitoring Method.

Another innovation is the adaptive classification threshold mechanism based on F1-score optimization. Most studies in this field rely on a fixed threshold (0.5), which is often suboptimal in real-world applications. F1-score optimization involves evaluating different threshold values and selecting the one that maximizes the F1-score on test or validation data. In this case, classification involves recognizing DNS queries as either legitimate (0) or malicious (1). The optimization parameter is the threshold ranging from 0 to 1, which converts probabilities (from the supervised model or autoencoder reconstruction error) into binary predictions. The method provides systematic threshold adjustment by computing the F1-score for different threshold values, automatically finding the optimal balance between precision and recall for malicious DNS query detection. System performance is evaluated using standard machine learning metrics, including F1, ROC-AUC, and the false positive rate.

For a comprehensive comparison of methods, multiple metrics are employed: precision, recall, F1-score, ROC-AUC, and PR-AUC. This approach provides a deeper understanding of each model's strengths and weaknesses and establishes an objective basis for decision-making in real-world applications.

To assess the method's effectiveness, a synthetic DNS query dataset was used, simulating various types of legitimate traffic and DNS tunneling with controlled characteristics. The synthetic data were used solely for method evaluation, not as part of the method itself. This creates an experimental environment for testing models in scenarios close to real-world conditions, where both legitimate and malicious queries exhibit high variability. This approach not only improves evaluation quality but also allows testing the robustness of the algorithms against spoofing attacks.

The block diagram illustrates the sequence of stages: preprocessing DNS queries, extracting statistical and structural features, parallel operation of the classifier and autoencoder, adaptive threshold adjustment, and final classification.

A key component of this method is the entropy function, which measures the randomness of a string. For a domain s, entropy is defined as:

$$H(s) = -\sum_{i=1}^{n} p_i \times \log_2 p_i \tag{1}$$

where p_i is the probability of a character i appearing in the string s, and n is the number of unique characters.

The model is evaluated using standard machine learning metrics. $Precision$ is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

where TP represents the number of correctly classified malicious examples, and FP represents the number of legitimate examples incorrectly classified as malicious.

$Recall$ is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where FN represents the number of malicious examples missed by the model.

The F1-score is the harmonic mean of precision and recall, calculated as:

$$F1 = 2\,\frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Additionally, integral classification quality metrics were computed, including the area under the ROC curve (ROC-AUC) and the area under the Precision-Recall curve (PR-AUC). The false positive rate (FPR) was also analyzed:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

where TN represents the number of correctly classified legitimate examples.

Characteristics of the Conducted Experiments

In this study, a synthetically generated dataset was used to simulate queries to the DNS from both legitimate and malicious sources. Legitimate queries included typical domain names, such as corporate services, Content Delivery Networks (CDNs), Universally Unique Identifiers (UUIDs), and random combinations of letters and digits simulating real user requests. Malicious examples were generated by encoding random strings using Base32 and Base64 encodings, hexadecimal representations, and hashing algorithms, corresponding to common DNS tunneling techniques. This data generation process resulted in a balanced dataset of 16,000 records (50% benign and 50% malicious).

The study also included high-entropy domain names that were not malicious but mimicked characteristic features of DNS tunneling to make the experiments more realistic and representative of real-world conditions. Specifically, some legitimate domain names contained random sequences of letters, long strings, and digits visually similar to malicious domains. Consequently, models needed to learn to distinguish not only simple patterns but also complex obfuscation patterns. This allowed us to assess the robustness of the TunnelEye approach against attacks employing both covert encoding and obfuscation imitation. The program was implemented in Python using the scikit-learn, TensorFlow, and matplotlib libraries. Statistical analysis employed string processing and traditional numerical properties, including domain name length, entropy, number of subdomain levels, and character ratios. Structural analysis relied on character n-gram features generated by CountVectorizer, ranging from 3 to 5 characters and limited to the top 3,000 most frequent characters.

For comparison, a baseline model using only statistical features and an autoencoder (a multilayer neural network consisting of a compression encoder and a symmetric decoder layer) were also employed. The baseline statistical model was used exclusively for comparative analysis, whereas TunnelEye implements a dynamic multi-level model capable of adapting to new traffic types.

All computations were performed in the Google Colab environment, enabling efficient use of computational resources for model training and easy replication of experimental results. The obtained metrics were used to compare the baseline statistical model, TunnelEye, and the autoencoder across different performance measures, confirming the scientific novelty of the method.

The supervised TunnelEye classifier and the autoencoder operate in parallel, each producing an independent anomaly score, while optimal decision thresholds are selected separately for each model using F1-score optimization. The autoencoder is not used as a direct decision fusion component but serves as an independent anomaly detector for comparative evaluation and robustness analysis against previously unseen attacks. The proposed architecture allows straightforward extension toward score-level or decision-

level fusion, which is left for future work. The results of the experiments are presented in the Results section.

RESULTS AND DISCUSSION

It is important to note that the method produces a binary outcome (anomalous DNS query detected or not detected). The metrics are used solely to quantitatively assess the quality of this decision and to compare different approaches. The results confirm that TunnelEye's effectiveness is not due to individual components alone but results from their coordinated integration. The autoencoder strengthens the model in scenarios where supervised classification loses sensitivity, while the adaptive threshold ensures stability in real-world conditions. Thus, the experimental study fully meets the stated objectives, and the results demonstrate the completeness and practical relevance of the proposed method.

Fig. 3 illustrates how precision and recall change with varying classification thresholds. This allows us to assess the balance between detecting all attacks (recall) and minimizing false positives (precision).

Results:

- TunnelEye: Average precision ≈ 0.99, recall ≈ 0.99
- Baseline model: Average precision ≈ 0.88, recall ≈ 0.90
- Autoencoder: Average precision ≈ 0.65, recall ≈ 0.48

Thus, TunnelEye shows minimal performance degradation across threshold values, demonstrating its stability. The baseline algorithm performs slightly worse, while the autoencoder exhibits fluctuations and is less balanced.

The F1-score (**Fig. 4**) summarizes precision and recall, reflecting model quality across different threshold values. This metric is critical for finding the optimal balance between errors and false positives.

Results:

- TunnelEye: Maximum F1 = 1.0 (threshold ≈ 0.06), average F1 ≈ 0.99
- Baseline model: Maximum F1 ≈ 0.86, average F1 ≈ 0.84
- Autoencoder: Maximum F1 ≈ 0.52, average F1 ≈ 0.49

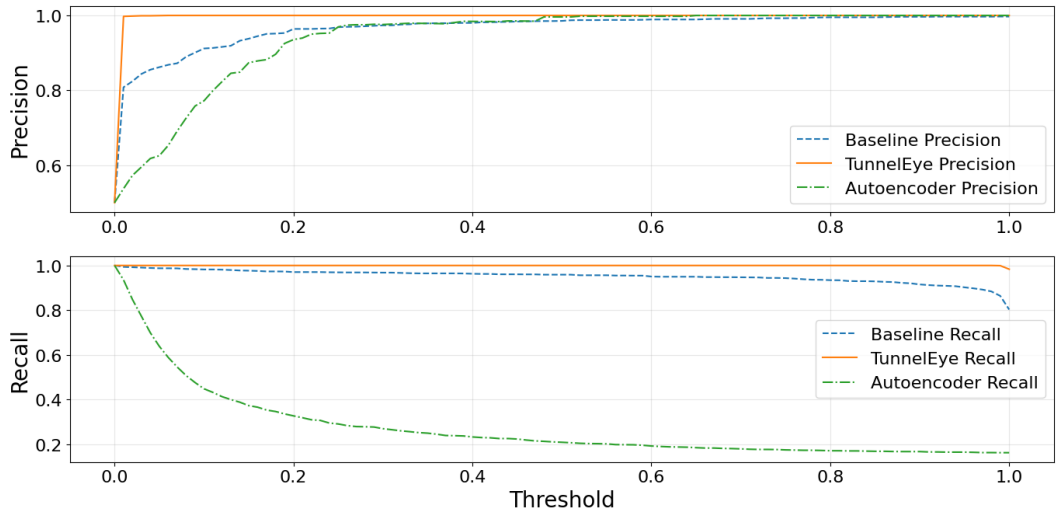

Fig. 3. Precision and Recall vs. Classification Threshold.

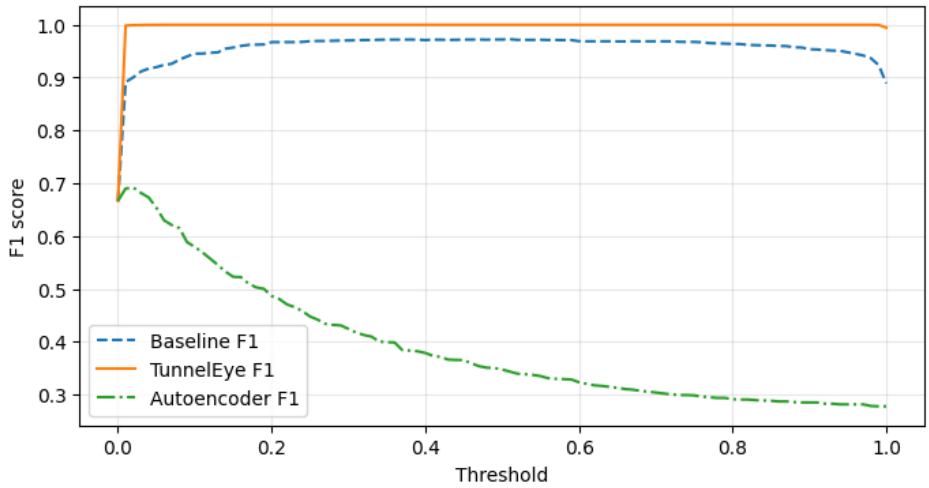

Fig. 4. F1-score vs. Classification Threshold.

TunnelEye significantly outperforms the other models, maintaining the highest overall performance. The baseline model is stable but not optimal, while the autoencoder performs substantially worse, particularly in terms of recall.

The ROC curve (**Fig. 5a**) measures the ratio of true positives to false positives, while the PR curve (**Fig. 5b**) shows the balance between precision and recall, which is particularly important for imbalanced classes. A large area under the curve (AUC) indicates a stable and effective model.

Results:

- TunnelEye: ROC-AUC = 1.0, PR-AUC = 1.0
- Baseline model: ROC-AUC ≈ 0.992, PR-AUC ≈ 0.991
- Autoencoder: ROC-AUC ≈ 0.722, PR-AUC ≈ 0.768

Thus, TunnelEye demonstrates the best performance, with the other models lagging behind.

Fig. 6 presents a “performance metric” that evaluates all models at the same threshold. This metric compares precision, recall, F1-score, ROC-AUC, PR-AUC, FPR, as well as the absolute values of TP, FP, TN, and FN.

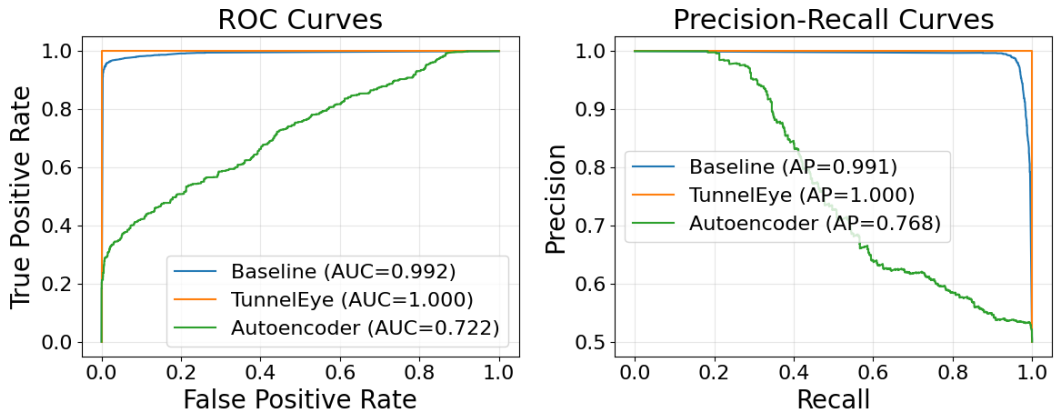

Fig. 5. ROC (a) and PR (b) curves.

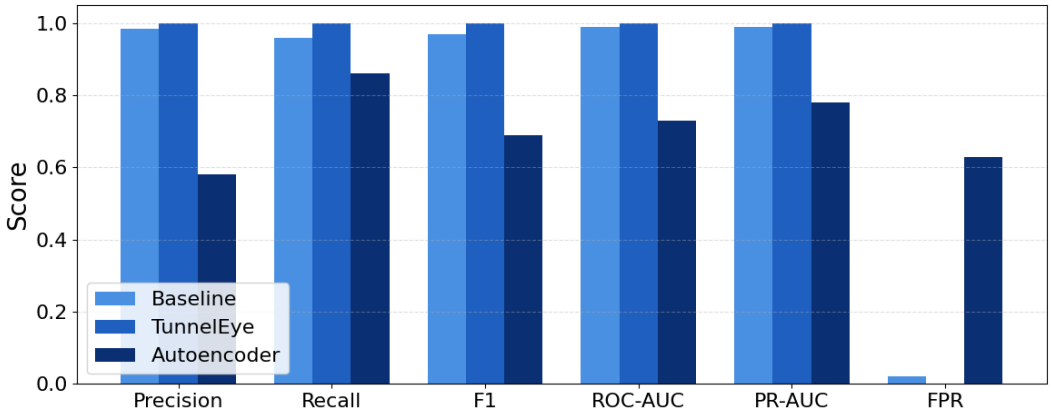

Fig. 6. Model Comparison at Fixed Threshold (0.5).

According to the comparative analysis (**Fig. 6**), TunnelEye achieved the best results (Precision, Recall, F1, ROC-AUC, PR-AUC = 1.0; FPR = 0), while the baseline model achieved moderate metrics (Precision ≈ 0.815, Recall ≈ 0.91, F1 ≈ 0.86, ROC-AUC ≈ 0.944, PR-AUC ≈ 0.936, FPR ≈ 0.197).

The autoencoder showed significantly lower metrics (Precision ≈ 0.72, Recall ≈ 0.41, F1 ≈ 0.52, ROC-AUC ≈ 0.65, PR-AUC ≈ 0.61, FPR ≈ 0.32), making it less suitable for standalone use. Therefore, TunnelEye can be considered an effective method for detecting DNS tunneling and similar attacks.

The experiments were conducted on a synthetically generated dataset with controlled characteristics. This approach allows for a fair comparison of models under identical conditions; however, it does not fully capture the complexity of real DNS traffic. To address this limitation, a public DNS dataset containing real domain names with naturally occurring feature distributions and inherent variability was used (**Fig. 7**). daumel/dns-tunneling-dataset is a public dataset containing DNS query traffic with tunneling, generated by various DNS tunneling tools.

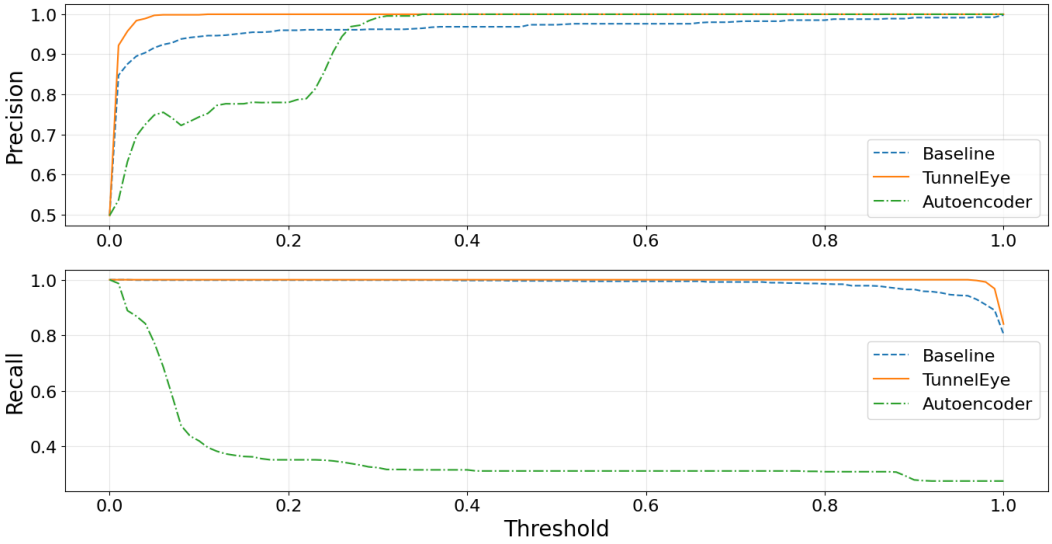

Fig. 7. Precision and recall versus classification threshold on the real DNS traffic dataset.

On this dataset, perfect metric values are not achieved; nevertheless, TunnelEye consistently demonstrates a better balance between precision and recall compared to baseline models. Therefore, the results confirm that the effectiveness of the proposed method is not a consequence of the artificial simplicity of synthetic data but is maintained under realistic DNS traffic conditions. Future studies will expand the experiments to other datasets to assess the method's performance across varied and realistic DNS traffic conditions.

Achieving ideal ROC-AUC and PR-AUC values in a controlled experiment should be interpreted as an indicator of the method's effectiveness under clearly separable class conditions, rather than as a universal measure of its behavior in real networks. Additional experiments on a real DNS dataset show that, in the presence of more complex mimicry scenarios and partial feature overlap between legitimate and malicious domains, the metrics become less ideal. Nevertheless, TunnelEye maintains an advantage over baseline statistical models and the autoencoder. This demonstrates the method's resilience to more sophisticated attacks and confirms that its effectiveness is not limited to synthetic scenarios.

To extend the comparative analysis, additional modern machine learning methods were also evaluated, including Logistic Regression and Linear SVM with character n-gram features, as well as Isolation Forest as a representative anomaly-based model. However, in the experiments, these methods did not demonstrate any advantage over TunnelEye in the context of DNS traffic analysis and were therefore not included in the corresponding plots.

CONCLUSION

In this paper, the TunnelEye monitoring method is proposed, which combines statistical methods, structural analysis, and an autoencoder for anomaly detection, demonstrating exceptional effectiveness in identifying malicious DNS queries. This combination allows the system to handle both known patterns and new, previously unseen malicious queries. Experimental studies show that TunnelEye achieves the highest precision and recall, significantly outperforming baseline statistical and autoencoder models. This demonstrates the effectiveness of a multi-stage approach and its applicability in real-world conditions with highly variable DNS traffic.

The scientific innovation lies in the method's ability to adaptively adjust classification thresholds and integrate multiple detection techniques for identifying malicious queries. This ensures high performance even under varying operational conditions, making it well-suited for integration into enterprise network traffic monitoring systems. Comparative analysis of different models demonstrates that TunnelEye not only provides high precision and recall but also reduces false positive rates, which is critical for practical deployment.

Experimental results indicate that TunnelEye achieves an average anomaly detection precision and recall close to 0.99, significantly surpassing both the baseline statistical model and the autoencoder. This highlights the advantage of the multi-stage method in detecting both known and novel malicious patterns. The F1-score further confirms TunnelEye's high performance (average F1 ≈ 0.99). These results demonstrate TunnelEye's reliability and its ability to minimize false positives. At a fixed threshold of 0.5, TunnelEye also showed high precision and recall. This comparison confirms that the proposed approach outperforms both traditional methods and autoencoders, achieving superior results across all metrics.

Overall, the comparative analysis demonstrates that TunnelEye is an effective method for detecting DNS tunnels and covert communication channels. Its multi-level adaptive approach ensures high accuracy, protects against zero-day attacks, and reduces false alarms, making it suitable for integration into corporate security and real-time network monitoring systems. TunnelEye can be considered an efficient and flexible

tool for detecting malicious DNS queries and DNS tunneling. Its multi-level adaptive design provides comprehensive network protection and opens avenues for further research in the detection of covert channels and anomalous traffic.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, writing, and publication of this article.

CONFLICT OF INTEREST

The author declares that the research was conducted in the absence of any.

AUTHOR CONTRIBUTIONS

The author has read and agreed to the published version of the manuscript.

REFERENCES

[1] Gonzalez Casanova, L. F., & Lin, P. C. (2021). Generalized classification of DNS over https traffic with deep learning. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Tokyo, Japan, 2021, pp. 1903–1907. https://ieeexplore.ieee.org/document/9689667

[2] Ichise, H., Jin, Y., & Iida, K. (2023). Policy-based detection and blocking system against abnormal applications by analyzing DNS traffic. In *2023 22nd International Symposium on Communications and Information Technologies (ISCIT)*, Sydney, Australia, 2023, pp. 1–6. https://doi.org/10.1109/ISCIT57293.2023.10376042

[3] Zhang, C., Hu, X., Pan, X., Cheng, G., Li, R., & Wu, H. (2025). Accurate and early detection of IoT malware via DNS traffic analysis with deep learning. In *ICC 2025 – IEEE International Conference on Communications, Montreal, QC*, Canada, 2025, pp. 2665–2670. https://doi.org/10.1109/ICC52391.2025.11161323

[4] Ganesh, N., Parihar, A. S., & Ghosh, G. (2023). Analysing network traffic and implementing diverse technologies to examine different components of the network. In *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, 2023, pp. 1–10. https://doi.org/10.1109/ICTBIG59752.2023.10456258

[5] Harishkumar, S., & Bhuvaneswaran, R. S. (2024). Unveiling domain generation algorithms in DNS log traffic: A next-generation intelligent framework for dynamic anomaly detection and mitigation through machine learning analysis. In *2024 15th International Conference on Computing Communication and Networking Technologies (*ICCCNT), Kamand, India, 2024, pp. 1–7. https://doi.org/10.1109/ICCCNT61001.2024.10726248

[6] Wu, X., Wang, X., Song, Y., & Ding, P. (2024). SSPT: A self supervised network traffic anomaly detection method. In *2024 20th International Conference on Mobility, Sensing and Networking (MSN)*, Harbin, China, 2024, pp. 1206–1207. https://doi.org/10.1109/MSN63567.2024.00177

[7] Zou, F., Ren, Y., Zhu, J., & Tang, J. (2021). Detecting data leakage in DNS traffic based on time series anomaly detection. In *2021 IEEE 23rd International Conference on High Performance Computing & Communications; 7th International Conference on Data Science & Systems; 19th International Conference on Smart City; 7th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, Haikou, Hainan, China, 2021, pp. 503–510. https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00090

[8] Du, X., et al. (2022). Design of an autoencoder-based anomaly detection for the DoH traffic system. In *2022 IEEE 25th International Conference on Computer Supported*

Cooperative Work in Design (CSCWD), Hangzhou, China, 2022, pp. 763–768. https://doi.org/10.1109/CSCWD54268.2022.9776029

[9] Hzami, M., Mahersia, H., & Bejaoui, T. (2025). Multi-level cyberbullying detection on social media using machine and deep learning models. In *2025 5th IEEE Middle East and North Africa Communications Conference (MENACOMM)*, Byblos, Lebanon, 2025, pp. 1–6. https://doi.org/10.1109/MENACOMM62946.2025.10911024

[10] Huang, X., Zhu, X., Xu, X., Zhu, M., Nian, A., & Guo, Y. (2022). Multi-granularity perceptual ensemble learning model with an application. In *2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*, Xiamen, China, 2022, pp. 234–242. https://doi.org/10.1109/MLCCIM55934.2022.00047

[11] Wang, B., Xiong, G., Gou, G., Song, J., Li, Z., & Yang, Q. (2023). Identifying DoH tunnel traffic using core features and machine learning method. In *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Rio de Janeiro, Brazil, 2023, pp. 814–819. https://doi.org/10.1109/CSCWD57460.2023.10152678

[12] Rana, S., & Aksoy, A. (2021). Automated fast-flux detection using machine learning and genetic algorithms. In *IEEE INFOCOM 2021 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, 2021, pp. 1–6. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484614

ОЦІНЮВАННЯ ПРОДУКТИВНОСТІ БАГАТОРІВНЕВОЇ МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ АНОМАЛЬНИХ DNS-ЗАПИТІВ

Андрій Сеник
Національний університет «Львівська політехніка»,
вул. Степана Бандери, 12, Львів, 79013, Україна

АНОТАЦІЯ

Вступ. У сучасних системах мережевої безпеки DNS-трафік дедалі частіше використовується для прихованої передачі даних і керування шкідливими системами (command-and-control). Існуючі методи машинного навчання нерідко мають обмежену здатність адаптації до нових шаблонів атак і стикаються з дисбалансом між точністю виявлення та рівнем хибнопозитивних спрацьовувань.

Матеріали та методи. Запропоновано багаторівневий метод виявлення зловмисних DNS-запитів TunnelEye, що інтегрує аналіз статистичних ознак, структурне n-грамне моделювання та виявлення аномалій. Для початкового розмежування легітимних і підозрілих запитів використовуються статистичні властивості доменних імен, зокрема довжина рядка, ентропія та співвідношення буквено-цифрових символів. Структурний аналіз на основі символьних n-грам дає змогу ідентифікувати локальні шаблони, пов'язані з кодуванням даних, таким як Base32 і Base64. Супервізований класифікатор TunnelEye та автокодер працюють паралельно, кожен із яких використовує незалежно оптимізований поріг на основі F1-міри для визначення аномальних DNS-запитів.

Результати. Експериментальна оцінка з використанням стандартних метрик машинного навчання (precision, recall, F1-score, ROC-AUC, PR-AUC та рівень хибнопозитивних спрацьовувань) показує, що TunnelEye стабільно перевершує базові статистичні моделі та окремі автокодери. Запропонований метод забезпечує високу точність і повноту виявлення за мінімального рівня хибнопозитивних спрацьовувань. Експериментальні результати показали, що метод TunnelEye

забезпечує середні значення precision, recall та F1-міри на рівні близько 0,99, перевищуючи базову статистичну модель більш ніж на 10% та істотно зменшуючи кількість хибнопозитивних спрацьовувань.

Висновки. TunnelEye пропонує комплексне та адаптивне рішення для виявлення зловмисних DNS-запитів шляхом поєднання керованого та некерованого навчання з динамічною оптимізацією порогів. Здатність методу збалансовувати точність виявлення та зменшення кількості хибнопозитивних спрацьовувань робить його придатним для впровадження в сучасні корпоративні системи кібербезпеки для моніторингу DNS-трафіку в реальному часі.

Ключові слова: DNS-трафік, виявлення аномалій, машинне навчання, багаторівнева модель

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
23 January, 2026	05 February, 2026	09 February , 2026	30 March, 2026