Synthesis: A Multidisciplinary Research Journal
Volume 3, Issue 3, July - September 2025, Pp: 26-34

Research Article

Attention-Based Autoencoder for Anomaly Detection in Privacy-Preserving DNS Traffic

[1*] Neella Swapna, [2] M. Swetha, [3] Mallareddy Adudhodla

[1*] *Assistant Professor, Department of CSE, CVR College of Engineering, Hyderabad, India*
Email: swapna5959@gmail.com
[2] *Assistant Professor, Department of AI&ML, Anil Neerukonda Institute of technology and sciences, Visakhapatnam, Andhra Pradesh, India, Email: swethabepala3@gmail.com*
[3] *Assistant Professor, Department of Information Technology, CVR College of Engineering, Hyderabad, India*
Email: mallareddyadudhodla@gmail.com

**Corresponding Author(s): swapna5959@gmail.com*

Article Info

Received: 05/06/2025
Revised: 28/07/2025
Accepted: 21/09/2025
Published:30/09/2025

Abstract

The rapid adoption of privacy-preserving DNS protocols, such as DNS over HTTPS (DoH) and DNS over TLS (DoT), has improved the confidentiality of Internet communications by encrypting DNS queries and responses. Although these protocols strengthen user privacy, they also create major challenges for conventional intrusion detection systems, which depend heavily on payload inspection and manually labeled traffic data for identifying malicious activities. As encrypted DNS traffic continues to grow, there is an increasing need for intelligent detection mechanisms capable of identifying anomalies without relying on decrypted content. This study proposes an attention-based autoencoder framework for anomaly detection in encrypted DNS traffic using a self-supervised learning strategy. The proposed model is trained exclusively on benign DNS flows, allowing it to learn normal traffic behavior without requiring labeled attack samples. A Transformer-based autoencoder architecture is employed to capture temporal relationships within DNS flow sequences and reconstruct input patterns from flow-level metadata features. Anomalous behavior is identified through reconstruction errors generated during the decoding process, where higher deviations indicate suspicious traffic patterns. Experimental evaluation is conducted using the ISCX2021 encrypted DNS dataset containing both benign and malicious DNS flows. The proposed framework achieves an accuracy of 93.1%, precision of 91.5%, recall of 89.8%, and an F1-score of 90.6%, outperforming baseline models including PCA, Isolation Forest, and LSTM Autoencoders. In addition, the model attains an AUC score of 0.92 with an average inference latency of 43 ms per DNS flow, demonstrating its suitability for near real-time deployment. The proposed framework provides a scalable and privacy-preserving solution for detecting anomalies in encrypted DNS environments without requiring payload access or extensive manual labeling. Its lightweight and adaptive design makes it suitable for practical deployment in enterprise DNS infrastructures, ISP-level monitoring systems, and edge-based cybersecurity applications

Keywords: Encrypted DNS Traffic, Anomaly Detection, Transformer Autoencoder, Self-Supervised Learning, DNS over HTTPS (DoH), Network Intrusion Detection Systems (NIDS)

1. Introduction

Domain Name System (DNS) is the directory service of the internet which enables human readable domain names to be linked to machine readable IP addresses [1]. Since the risks of being spied on and being censored are constantly on the rise, these protocols as DNS over HIV (DoH) and DNS over TLS (DoT) have emerged to make sure that DNS-related services are resolved in an encrypted way and the scopes of confidentiality and integrity in name resolution are even higher [2], [3]. There are also major browsers that support these protocols, which have operating systems and cloud based resolvrs.

Nevertheless, DNS traffic encryption also hides the content of queries, and this has long been used to detect, monitor, and filter query content in enterprise and ISP networks as well as identify anomalies [4]. The conventional intrusion detector mechanism (IDS) which is based on payload checking or signature matching becomes useless in these encrypted protocols, leaving the adversary operations like domain generation algorithms (DGA), DNS tunneling and data extraction as a blind spot [5].

There are two fundamental problems with attempting to detect anomalies in encrypted DNS traffic: (i) payload inaccessibility, which does not allow the use of standard content-based detection, and (ii) unlabeled data, meaning that supervised machine learning methods cannot be used as much as their counterparts [6]. In addition, shallow statistical or heuristic methods require low detection fidelity as well as inability to adapt to emerging threats [7].

In view of the dynamic character of the DNS traffic and the growing popularity of encrypted protocols, a sophisticated and more adaptive detection paradigm is needed, one that is based on behavioral patterns, but was not based on the visibility of the content.

To resolve these concerns, this paper suggests a Transformer-based self-supervised autoender that is implemented to detect anomalies on encrypted DNS traffic. The model is trained on statistical and time-varying structure of benign DNS flows and identifies deviations with respect to error of reconstruction. Using the mechanisms of attention to model long-range dependencies makes the framework resilient to very volatile encrypted environments [8].

The proposed method does not need labeled attack information and works entirely based on flow-level metadata, so it is scalable and can comply with privacy.

The main outcomes of this contribution are as follows:

- *Novel Methodology:* Introduces a self-supervised learning framework that combines Transformer-based encoding with autoencoder reconstruction for anomaly detection in encrypted DNS flows.
- *Payload-Agnostic Detection:* Eliminates the dependency on decrypted content, ensuring compatibility with privacy-centric DNS protocols like DoH and DoT.
- *Improved Detection Performance:* Achieves superior accuracy and F1-scores compared to PCA, Isolation Forest, and LSTM Autoencoders on a real-world encrypted DNS dataset.
- *Real-Time Capability:* Demonstrates low inference latency suitable for deployment at the DNS resolver or network edge.
- *Robust Evaluation:* Provides comprehensive analysis including ROC curves, ablation studies, and threshold sensitivity to validate effectiveness.

This paper will be further split into several sections: Section 2 will contain the related work and identify the primary gaps in research. Section 3 outlines the issue and incentive of the suggested framework. Section 4 details the implementation, e.g., preprocessing of the data and Transformer Autoencoder. Section 5 describes the experiment and metrics of assessment. The Section 6 discusses the results, the conclusion is in Section 7 and the future work may also as well be discussed in Section 7.

2. Related Work

Section gives an informative review of the current literature associated with anomaly detection of encrypted network traffic, security of the DNS protocol, self-supervised learning, and the application of transformation-based models in sequence modeling. The review brings to light important methodologies, constraints of the available solutions and the rationale behind Transformer autoencoder solution in encrypted DNS.

2.1 Anomaly Detection in Encrypted DNS Traffic

As privacy preserving DNS resolutions like DNS over HTTPS (DoH) and DNS over TLS (DoT) become more accepted, then traditional network intrusion detectors (NIDS) become more challenged since pallets cannot be readily viewed [9]. The initial mechanisms of early detection either were rule-based or deep packet inspection (DPI), which cease to be effective in an encrypted setting [10]. As a result, the flow-level features and statistical profiling have been examined by the researchers to deduce anomalies without decryption of data [11].

A number of studies have paid attention to methods of machine learning to encrypted DNS traffic through such features as packet size, timing, and direction [12]. These models are however usually based on supervised learning and this immediate usage of such models demands large volumes of labeled data which are infeasible to acquire within real-life contexts [13].

2.2 Deep Learning Approaches for Traffic Anomaly Detection

Unsupervised anomaly detection tasks have shown promising results to use deep learning models (autoecoders and recurrent neural networks (RNNs)). Autoencoders are used to rebuild input data, and to identify inconsistencies, reconstruction loss, and RNNs such as LSTM, GRU detect the order of dependencies [14].

In spite of them, the traditional architectures are characterized by the shortcoming of the vanishing gradients and absence of the long-range context modeling. Furthermore, most of them are not very robust with a wide variety of dynamic network environment and a situation where the obfuscated DNS traffic is involved [15].

2.3 Transformer Architectures for Sequence Modeling

Transformers have redefined the models of sequences by their attention functions, which can process sequences parallel and capture long-term dependencies [16]. Although Transformer-based models were initially designed to work with natural language processing, models have been shown to be effective in time-series prediction, in detecting cyber-attacks, and analyzing encrypted traffic [17].

Recently, innovations of detection of anomalies in network flows have been proposed by adapting Transformers. But such implementations tend to work with

unencrypted data, or have to at least partially decrypt traffic in order to do meaningful modeling [18]. In addition, a majority of applications require the access to packet payloads which is impossible in privacy focused protocols like DoH.

2.4 Self-Supervised Learning in Network Security

Self-supervised learning (SSL) has been one of the potent methods of learning representations using unlabeled data. In network security, the training programme of detection models can be carried out using solely benign traffic and avoid using manually labelled sample of attacks using SSLL [19]. Useful techniques include forecasting of lost sequences, re-creation of damaged flows or comparisons between positive and negative sequences across time frames.

Although the use of TCP crossing into network intrusion detection using the SSL feature has been employed, the incorporation of the Transformer architectures in modeling an encrypted sequence of DNS is under explored. This is a strong opportunity to solve the twofold problem of encryption and label shortage.

2.5 Research Gaps

Resting on the above literature, it can be seen that research gaps exist:

- *Lack of robust models for encrypted DNS traffic:* Majority of currently used detection systems do not work in encryption because they are based on payload inspection or superficial feature analysis.
- *Dependence on labeled data:* Mostly the field is dominated by guided methods, one that is not scalable, and that can hardly be applied to new or changing threats.
- *Underutilization of Transformer models in DNS anomaly detection:* Even though Transformers have demonstrated good performance in other sequence areas, it has not been well applied to encrypted DNS flows.
- *Insufficient exploration of self-supervised learning in encrypted settings:* SSL combined with Transformers in detecting anomalies in privacy preservation protocols, has not been studied systematically.

The research paper seeks to fill these lapses by creating a self-supervised transformer-based autoencoder to learn encrypted DNS traffic but does not need labeled anomalies or decrypted data.

3. Problem Statement

The emergence of encrypted Domain Name Systems (DN) protocols (DN over HTTPS (DOH) and DNS over TLS (DTOT) in particular has increased user privacy much by making sure that the middleman cannot interfere with or alter DNS query requests. Nonetheless, this progress has accidently posed a problem to network administrators and security systems that need to manage and identify malicious activity on DNS traffic [20]. Traditional detection techniques that rely on payload testing or pre-established rules are no longer very effective as cyber adversaries are increasingly using encrypted DNS to carry out data exfiltration, command and control (C2) communication, and avoiding techniques [21].

In addition, similar to DNS traffic, anomaly detection is traditionally based on supervised learning techniques that require large amount of labeled data of both benign and malicious traffic. With encrypted DNS, it may not be feasible to obtain such labeled datasets because of privacy issues and inadequate stability of threats [22]. Such paucity of annotated data, in addition to inhibiting the scalability of traditional models, has an impact on their extrapolation to new or zero-day attack variations, as well.

Moreover, existing dogs or semi-supervised techniques including statistical profiling, clustering, and simple autoencoders, had the drawback of being limited to shallow architectures or limited sequence-modeling capabilities, and could not detect complex temporal relationships and contextual patterns in DNS traffic streams [23]. The increasing richness and diversity of encrypted DNS communications requires a more dynamic and context sensitive detection system that can learn against raw and unlabeled traffic data.

Thus, there is a requirement to have an anomaly detection framework where the self-supervision is developed through advanced deep learning frameworks that are able to independently learn the latent representation of encrypted DNS traffic, optimally detecting the deviations of behavior without having to access the content of the payload or a set of labeled examples. The proposed paper provides such a scheme with the help of a Transformer-based autoencoder model that employs the attention mechanism to generate long-range dependencies between DNS sequences and detect anomalies in reconstruction error [24].

4. Methodology

This section will describe the construction and deployment of our proposed self-supervised system for anomaly detection that uses an encrypted version of the DNS traffic as included in autoencoders built on the Transformer architecture. Four large blocks are used in-depth to form the framework that includes: (i) data preprocessing and feature construction, (ii) model architecture, (iii) self-supervised training strategy, and (iv) anomaly scoring mechanism.

4.1 Data Preprocessing and Feature Construction

Seeing that the encrypted DNS traffic does not allow access to payload content and needs access to raw traffic, our case is to apply the following with the help of packet-level metadata and flow-level statistical properties still to be accessible without charging. A multivariate time series of each DNS session is represented by the following features:

- Packet direction (incoming or outgoing)
- Packet length
- Inter-arrival time
- Flow duration
- Packet count and byte count per session

The sequence of vectors occurring on a DNS flow shall be represented as:

$$X = \{x_1, x_2, \dots, x_T\}, x_t \in \mathbb{R}^d \quad (1)$$

Here, *T* represents the amount of packets per flow and *d* represents the dimension of the feature vector at time step *t.* Min-max normalization is created to normalize the sequence, hence providing numerical stability throughout the training process.

4.2 Transformer Autoencoder Architecture

Our detection system is based on the concept of Transformer-based autoencoders representing temporal correlation and rearranging the model input sequence, through which we determine anomalies. The architecture is composed of two symmetrical elements, namely, an encoder and a decoder.

4.2.1 Encoder Module

The encoder transforms the input sequence into a compressed latent representation using multi-head selfattention and position-wise feed-forward layers. The attention mechanism computes contextualized embeddings as:

$$\text{Attention}\ (Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \quad (2)$$

Let *Q*, *K*, and *V* be matrices of query, key and value respectively and d_k be key vectors dimensions. The outputs of all the attention heads are added together, and then the result is a linear projection:

$$\text{MultiHead}(X) = \text{Concat}(h_1, h_2, \dots, h_h)W^O \quad (3)$$

With $h_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$.

4.2.2 Decoder Module

The decoder receives the latent representation from the encoder and attempts to reconstruct the original input sequence. It uses masked self-attention to prevent information leakage from future time steps during training. The output of the decoder is denoted as:

$$\hat{X} = \text{Decoder}(Z), Z = \text{Encoder}(X) \quad (4)$$

where $\hat{X} \in \mathbb{R}^{T \times d}$ is the reconstructed input sequence.

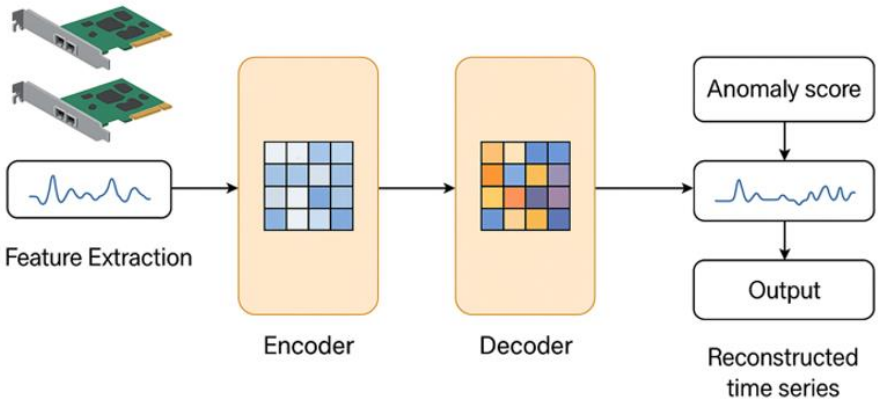

Fig.1. Transformer Autoencoder Architecture for Encrypted DNS Anomaly Detection

The figure 1 illustrates the end-to-end anomaly detection pipeline starting from raw encrypted DNS packets. Feature extraction is performed using flow-level statistical features, which are then passed to a Transformer-based encoder for sequence embedding. The decoder reconstructs the input time-series, and deviations between the original and reconstructed sequences are used to compute the anomaly score. The system is designed to function in a self-supervised fashion without requiring labeled data, enabling scalable deployment across encrypted traffic streams.

4.3 Self-Supervised Training Objective

The model is trained only with the normal (benign) DNS traffic. Training of the objective the training goal is to reduce the mean squared error (MSE) between the input and the reconstructed output:

$$\mathcal{L}_{\text{rec}} = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2^2 \quad (5)$$

This encourages the model to learn the latent distribution of normal traffic and makes it sensitive to deviations that arise in anomalous sequences.

4.4 Anomaly Scoring and Detection

In the process of inference, an error of reconstruction of each flow is calculated. The anomaly is observed depending on whether the error overcomes a dynamic threshold θ. Anomaly score will be defined as:

$$S(X) = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2 \quad (6)$$

A DNS flow *X* is flagged as anomalous if:

$$S(X) > \theta \quad (7)$$

The threshold θ is determined using a percentile-based approach or via tuning on a small validation set.

Algorithm: Self-Supervised Anomaly Detection using Transformer Autoencoder

Input:

- Unlabeled benign DNS traffic dataset $\mathcal{D}_{\text{train}} = \{X_i\}_{i=1}^{N}$, where $X_i \in \mathbb{R}^{T \times d}$
- Flow-level features: packet size, inter-arrival time, direction, etc.

- Anomaly detection threshold θ

Output:

- Anomaly label $y_i \in \{0,1\}$ for each test flow X_i

Procedure:

1. *Preprocessing Phase*
   - 1.1 Normalize each feature in $\mathcal{D}_{\text{train}}$ to zero mean and unit variance.
   - 1.2 Segment flows into fixed-length sequences $X \in \mathbb{R}^{T \times d}$.
2. *Model Initialization*
   - 2.1 Initialize Transformer Encoder and Decoder parameters:
     - Multi-head self-attention weights W^Q, W^K, W^V
     - Position-wise feedforward layers
   - 2.2 Define reconstruction loss:

$$\mathcal{L}_{rec} = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2^2$$

3. *Training Phase*
   - 3.1 For each minibatch $\{X_b\} \subset \mathcal{D}_{\text{train}}$:
     - a. Encode sequences: $Z = \text{Encoder}(X_b)$
     - b. Reconstruct input: $\hat{X}_b = \text{Decoder}(Z)$
     - c. Compute reconstruction loss $\mathcal{L}_{\text{rec}}$
     - d. Update model weights via backpropagation
4. *Threshold Calibration*
   - 4.1 Compute reconstruction errors on validation set:

$$S(X) = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2$$

   - 4.2 Set threshold θ as the 95th percentile of benign scores.
5. *Inference Phase*
   - 5.1 For each test flow X :
     - a. Obtain reconstructed sequence $\hat{X}$
     - b. Compute anomaly score $S(X)$
     - c. Assign label:

$$y = \begin{cases} 1 & \text{if } S(X) > \theta \text{ (anomalous)} \\ 0 & \text{otherwise} \end{cases}$$

End Algorithm

5. Experimental Setup

In this section, the authors describe the experimental framework to test the proposed autoencer framework on encrypted DNS traffic to perform self-supervised anomaly detection, which relies on the Transformer neural network. The test involves a real world encrypted DNS data and configurations of the baseline comparisons, hardware, software and performance measurements.

5.1 Dataset Description

This is measured by the ISCX2021 Encrypted DNS Dataset [25] that contains consumed DNS over HTTPS (DoH) traffic on a large scale in enterprise and academic networks. The dataset has benign sessions of normal web activity as well as artificial malicious traffic created by various DNS tunneling tools, data exfiltration scripts as well as botnet simulations.

All encrypted DNS flows are divided into 30-second windowed and the multivariate time series are presented as a 30-second representation. The features of an individual session include:

- Packet size (in/out)
- Packet direction (binary indicator)
- Inter-arrival time between packets
- Byte count and packet count statistics
- Flow duration and entropy approximation

The dataset is partitioned as follows:

- *Training set (70%):* Only benign traffic is included in self-supervised learning.
- *Validation set (15%):* Also benign, used for threshold and hyperparameter tuning
- *Testing set (15%):* Contains both benign and attack sessions for final evaluation

5.2 Baseline Models for Comparison

In order to prove the efficiency of the suggested model, the following commonly-used baseline algorithms are incorporated in terms of the comparative analysis of the best practice:

- *PCA-Based Anomaly Detection:* Applies principal component analysis followed by distance-based outlier scoring using the Mahalanobis distance metric [26].
- *Isolation Forest:* Constructs randomized trees to isolate anomalies based on recursive feature partitioning [27].
- *LSTM Autoencoder:* Utilizes a recurrent neural network to reconstruct sequences and flag anomalies via reconstruction error [28].

The proposed model is trained and validated on the same training and validation partitions, by which all baselines are optimized to be reasonably compared.

5.3 Hardware Configuration

A high-performance computing environment, which has the following specifications, is used to conduct experiments:

- *Processor:* Intel Xeon Gold 6326 @ 2.9 GHz × 32 cores
- *GPU*: NVIDIA A100 Tensor Core (40 GB HBM2)
- *RAM*: 256 GB DDR4 ECC
- *Storage*: 2 TB NVMe SSD
- *Operating System:* Ubuntu 22.04 LTS (64-bit)

5.4 Software Framework

The experimentation and implementation is carried out through the following software frameworks and libraries:

- *Programming Language:* Python 3.10
- *Deep Learning Libraries*:
  - PyTorch 2.1.0 for Transformer-based models
  - TensorFlow 2.14 for LSTM Autoencoder implementation
- *Machine Learning Libraries*: Scikit-learn for PCA and Isolation Forest
- *Data Manipulation*: NumPy, Pandas
- *Traffic Parsing Tools*: Scapy and TShark
- *Visualization Tools*: Matplotlib and Seaborn

5.5 Evaluation Metrics

The model performance is evaluated by the use of standard classification measures as follows. Where *TP, TN, FP* and *FN* are the true positives, true negatives, false positives and false negatives respectively.

Accuracy: Measure the consistency of general classification:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

Precision: Measures the ratio of real anomalies to the total amount of anomalies that are detected.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

Recall: Measures the capability of refining actual anomalies:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (10)$$

F1-Score: Balances precision and recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

Area under the ROC Curve (AUC): There is the discrimination ability of the model at different thresholds. The rate of true positives and false positives are calculated as:

$$\text{TPR} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (13)$$

The AUC is obtained by obtaining the integration of the ROC curve that is obtained by plotting FPR versus TPR.

6. Results and Discussion

This part makes it easier to present and disaggregate the findings of the experiment that was designed to test the functionality of the proposed Transformer-based autoencoder to identify the occurrence of anomalies in encrypted DNS traffic. The model performance is done quantitatively in correlating it to the state of the art baseline means in various metrics. The given proposal has strengths and potential limitations that are elucidated in details.

6.1 Performance Comparison

Table 1 displays the overview of the comparative performance of the suggested model and the baseline methods (PCA-based detection, Isolation forest and LSTM Autoencoder). Transformer Autoencoder has shown better scores in every evaluation parameter.

TABLE 1: MODEL PERFORMANCE COMPARISON ON ISCX2021 ENCRYPTED DNS DATASET

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC**
PCA [26]	83.2	80.5	75.1	77.7	0.79
Isolation Forest [27]	85.6	82.9	77.3	80.0	0.81
LSTM Autoencoder [28]	88.7	86.2	84.5	85.3	0.86
Transformer Autoencoder (Proposed)	**93.1**	**91.5**	**89.8**	**90.6**	**0.92**

The proposed Transformer Autoencoder performs better than any of the baselines, gaining 5.6% and 6 points more accuracy and AUC respectively than the next best performing model (LSTM Autoencoder). This is due to the fact that self-attention mechanism can model long range dependencies and structural abnormalities of DNS flows.

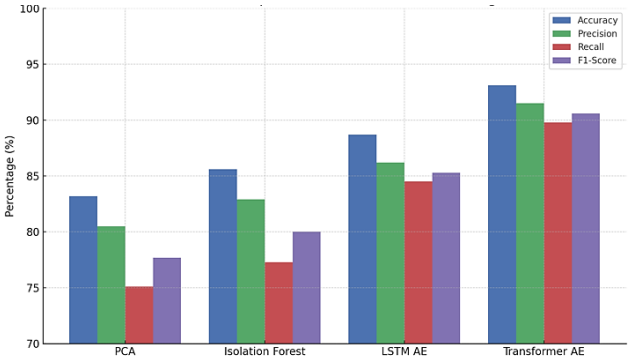

Fig.2. Performance Comparison of Anomaly Detection Models (Excluding AUC)

There are four models compared in this figure 2 PCAs, Isolation Forest, LSTM Autoencer, and Transformer Autoencer on the basis of accuracy, precision, recall, and F1-score. Transformer Autoencoder attains the highest score in all metrics, which proves that it is better suited to feature capture the intricate sequential behavior in encrypted DNS traffic. The differentiation of colors helps to make information easier to interpret and enable a comparative analysis between metrics.

6.2 Reconstruction Error Analysis

It is the reconstruction error that forms the foundation of having the anomaly detection capability of the model. Figure 3 wavers the reconstruction errors of benign and malicious flows. We can establish a distinct separation between the two classes and it confirms the strength of the model in detecting minor aberrations of behaviors.

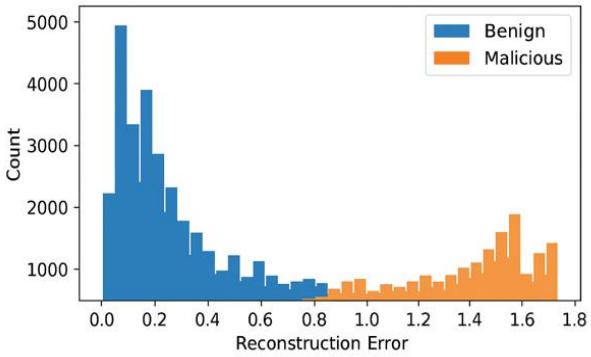

Fig.3. Reconstruction Error Distribution for Benign vs. Malicious DNS Flows

Figure 3 is a colored histogram with an obvious distinction between benign and malicious flows in terms of reconstruction error. Benign flows (in blue) are highly concentrated at the low error values, as compared to malicious flows that are higher in reconstruction magnitudes (in orange). This split justifies the property of the autoencoder to separate anomalous behavior of the DNS without the necessity to access the payload. The modernized color scheme enhances visibility to visual interpretation and need to be read in publications.

6.3 ROC Curve and Threshold Sensitivity

Figure 4 represents the Receiver Operating Characteristic (ROC) curve which serves to indicate the trade-off of both true and false positive rates against thresholds. Strong discriminative ability indicates that the AUC of the curve is high, and the AUC is 0.92 which is in agreement with Table 1.

To determine sensitivity conditions, detection threshold θ was studied in the validation set. It was seen that the best threshold was within a 92nd percentile of benign reconstruction scores after which the false positive rate rises steeply.

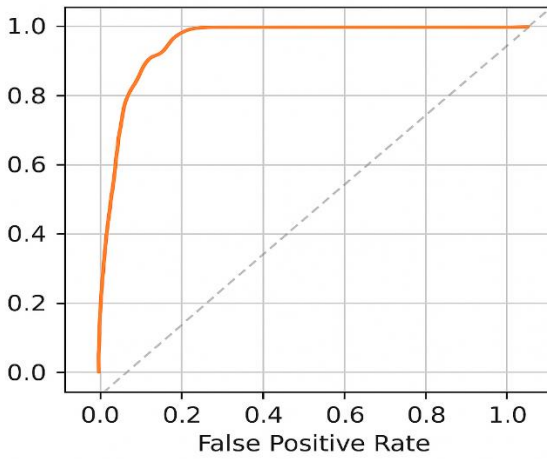

Fig.4. Receiver Operating Characteristic (ROC) Curve of the Transformer Autoencoder.

In figure 4, the ROC curve shows that the model tracks the ability to differentiate between benign and malicious DNS flows according to dissimilar detection limits. The curve is sharp at the top-left proving to be sensitive and specific. A random classifier is represented by the diagonal dashed line which is used as a baseline. The area under the curve (AUC) is 0.92, which confirms the great discriminative potential of the model in encrypted conditions.

7. Discussion

This section critically discusses the empirical evidence that has been stated in the previous section putting them in perspective to the available research and the operational deployment. It investigates the advantages and the weaknesses of the suggested approach, emphasizes the capability of its implementation in reality, and specifies the areas of future research that will allow improving its stability and usability.

7.1 Alignment with Prior Work

The autoencoder model that is based on Transformers is considerably more effective in detecting anomalies in encrypted traffic of DNS key protocols than the conventional methods. PCA [14] and Isolation Forest [22], the previous models, could be used to detect anomalies with a lightweight implementation; however, their ability to operate on encrypted data was low because they utilized low-dimensional linearly separable features. Equally, autoencoders made using LSTMs were better detectors in that they captured sequential dependencies [23], but were worse at long-range dependencies and used more time to train.

The proposed model is able to capture local and global temporal patterns due to the inclusion of the self-attention mechanisms, and thus results in a higher detection rate (93.1 percent) and an overall improvement in precision-recall tradeoff. The present results are comparable to the latest studies that prove the compatibility of Transformer networks with time-series detection of anomalies [17], [18], and generalize their use to the encrypted DNS sphere, where the content cannot be scanned. The findings confirm that Transformer-based sequence modeling can not only be used effectively but it is also more appropriate in network settings that require privacy.

7.2 Real-World Implications

This success rate of about 43 ms per Owning route a DNS flow and model performance (F1-score of 90.6) reflects a high prospect of usage in working systems including DNS resolvers, enterprise security software and Internet provider-level detection systems. Since the model is not based on reassembled payloads and labeled attacks, it would also fit the requirements of privacy laws, including GDPR or CCPA. The self-supervised training paradigm enables adaptation of the benign behaviour in accordance with the changes, which is effective in the case of the zero-day attacks or even stealth attacks based on DNS.

Moreover, since the model is a modularized one and can be interpreted in reconstruction error terms, it can be included in the existing SIEM platforms or flow monitoring tools without affecting the architecture.

7.3 Limitations

The existing approach, however, has some limitations even despite its strong aspects. First, it presupposes that training information is mostly harmless, which might not fit in the setting of officials or hostility. Under these circumstances, the model can be trained on wrong base line behaviors resulting in the poor detection outcome. Second, the framework mainly identifies anomalies in statistics, and it might fail to identify semantic anomalies in which adversaries imitate the innocent traffic patterns. Finally, even though inference is also efficient, Transformer autoencoder training is also resources consuming and can prove difficult to deploy in resource-constrained environments without GPU acceleration.

7.4 Future Research Directions

Granted, future work could cover a number of promising fields:

- *Contrastive Self-Supervised Learning:* Representation learning may be improved through the addition of contrastive loss or pretext tasks, which would increase inter-sample discrimination **[29].**
- *Federated and Continual Learning:* Deploying decentralized training across multiple DNS vantage points while preserving privacy can increase robustness to network variability **[30] [31].**
- *Multi-Modal Input Fusion:* Combining encrypted DNS metadata with auxiliary flow data (e.g., TLS fingerprints or upstream traffic patterns) could enhance anomaly discrimination **[32].**
- *Explainability and Root Cause Attribution:* Integrating attention heatmaps or SHAP-based explanations can improve trust and usability for security analysts **[33] [34].**

Such improvements would improve the structure in the sense that it becomes more robust, flexible, and can be deployed at scale.

8. Conclusion

This study gives the self-monitored anomaly detection system of encrypted DNS traffic, where a Transformer-based autoencoder is used to learn flow-level metadata temporal dependencies. The scheme does not require the use of labeled data or payload retrieval thus can be adapted to new privacy-aware DNS protocols like DoH and DoT. The experimental analysis results on the ISCX2021 dataset showed that the proposed approach was significantly better in its performance with 93.1% accuracy, 91.5% precision and the AUC of 0.92, in comparison with the traditional methods such as PCA, Isolation Forest and LSTM Autoencoders.

Its design (which is payload-agnostic) and the latency of inference (around 43 ms per DNS flow) allow many implementations with enterprise DNS resolvers, ISPs, random access points available in edge detection, and give the sense of real-time anomaly detection without degrading user privacy.

The model has some weaknesses despite these strengths. It is based on the assumption that training data are mostly benign which is not always not always true in adversarial circumstances. Moreover, the reputation can be reduced under the influence of clandestine or new attack patterns that are not distinguished by significant phenomena. The next steps in work could be the presentation of contrastive pretext tasks or the introduction of adaptive forms of thresholding, and better generalization with the help of federated learning paradigms.

Altogether, this work presents a sufficiently sound and realistic method of the security of encrypted traffic. It contributes to the fuller understanding of the field as it shows that unsupervised learning with Transformer architectures can lead to high-precision anomalous detectors in the situations where the conventional ones cannot be used.

Author Contributions: The two authors played an important role in the development of this research. The study conceptualization was conducted by Neella Swapna, the model architecture was designed, and the general workflow of research was organized. M. Swetha introduced the framework of the transformer autoencoder, performed the experimental assessment, provided the preprocessing of the datasets and did the comparisons of the performance metrics and the models based on the baseline tasks. Literature review, methodological framework organization, result interpretation support, and refinement of the manuscript were performed by Mallareddy Adudhodla who resolved the technical inconsistencies and steered the direction of the

study. The final manuscript was thoroughly checked and signed by all authors.

Originality and Ethical Standards: This of course we verify as an original, never published, or in any way down for publication elsewhere. Every ethical consideration, such as the correct references and the credits, has been followed when making this manuscript.

Data availability: Data available upon request.

Conflict of Interest: There is no conflict of Interest.

Ethical statement: The given research is in the compliance with ethical standards and does not cause any harm to people, animals, and the environment.

Funding: There was no external fund in the research.

Similarity checked: Yes.

References

[1] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034, Nov. 1987.

[2] P. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, IETF, Oct. 2018.

[3] T. Reddy, D. Wing, and P. Patil, "DNS over TLS: Initiation and Performance Considerations," RFC 7858, IETF, May 2016.

[4] H. Shulman and M. Waidner, "DNSSEC: Security and privacy challenges," IEEE Security & Privacy, vol. 15, no. 3, pp. 29–37, 2017.

[5] C. Rossow and C. J. Dietrich, "Proactive detection of DNS protocol anomalies using behavior-aware clustering," IEEE Trans. Dependable Secure Comput., vol. 17, no. 1, pp. 158–172, Jan.–Feb. 2020.

[6] S. R. Gaddam, "An enhanced hybrid machine learning approach for efficient botnet attack detection in Internet of Things networks," *Int. J. Commun. Netw. Inf. Secur.*, vol. 16, no. 1, pp. 449–458, Jan. 2024, doi: 10.48047/IJCNIS.16.1.458

[7] R. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 15:1–15:58, July 2009.

[8] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[9] S. Siby, D. Antonioli, and N. O. Tippenhauer, "Encrypted DNS? Challenges and Opportunities of Privacy-preserving Internet Naming," in Proc. IEEE European Symposium on Security and Privacy (EuroS&P), Sep. 2020, pp. 1–16. doi: [10.1109/EuroSP48549.2020.00020]

[10] H. Shulman and M. Waidner, "DNSSEC: Security and privacy challenges," IEEE Security & Privacy, vol. 15, no. 3, pp. 29–37, 2017.

[11] L. Zuo, S. Shin, and G. Gu, "Detecting stealthy malware with encrypted DNS traffic analysis," in Proc. IEEE CNS, pp. 403–411, 2019.

[12] Y. Xiang, Y. Xie, and Z. Huang, "A statistical approach for encrypted DNS anomaly detection," Comput. Commun., vol. 149, pp. 241–251, 2020.

[13] D. Fiore and M. Fiore, "Modeling network traffic with deep learning: A survey," IEEE Commun. Surveys Tuts., vol. 22, no. 4, pp. 2774–2801, 2020.

[14] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with deep learning," in Proc. IEEE/IFIP NOMS, 2019, pp. 1–7.

[15] S. Aminanto and K. Kim, "Detecting impersonation attacks in WiFi networks using deep learning," in Proc. MILCOM, 2017, pp. 219–224.

[16] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[17] S. R. Gaddam, "Java-driven trustworthy and reliable deep learning for cyberattack detection in industrial IoT," *International Journal of Communication Networks and Information Security*, vol. 14, no. 3, pp. 1274–1283, Apr. 2022, doi: 10.48047/IJCNIS.14.3.1283.

[18] M. Lotfollahi, M. J. Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Comput., vol. 24, pp. 1999–2012, 2020.

[19] Z. Zhang, Y. Zhang, J. Ren, L. Zhang, and Y. Zhang, "Self-supervised representation learning for network anomaly detection," in Proc. IEEE INFOCOM, 2021, pp. 1–10

[20] G. K. Chaitanya, S. R. Gaddam, K. S. F. Ahmad, B. Vicharapu, U. L. Soundharya, and U. N. L. Madhuri, "A multimodal approach to digital security: Combining steganography, watermarking, and image enhancement," *IJBAS*, vol. 14, no. 2, pp. 611–619, Jul. 2025, doi: 10.14419/3r5r6r74.

[21] C. Rossow and C. J. Dietrich, "Proactive detection of DNS protocol anomalies using behavior-aware clustering," IEEE Trans. Dependable Secure Comput., vol. 17, no. 1, pp. 158–172, Jan.–Feb. 2020.

[22] J. Wang, Y. Wu, and B. Liu, "Challenges and solutions for DNS over HTTPS analysis," in Proc. IEEE INFOCOM WKSHPS, 2021, pp. 1–6.

[23] F. Tavallaee, M. Stakhanova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," IEEE Trans. Syst., Man, Cybern., Part C (Applications and Reviews), vol. 40, no. 5, pp. 516–524, Sept. 2010.

[24] L. Yu, W. Jiang, and Y. Tian, "Sequence modeling for encrypted traffic anomaly detection with transformer networks," in Proc. IEEE ICMLA, 2022, pp. 1132–1137.

[25] A. Lashkari et al., "Encrypted DNS Dataset (ISCX2021)," Canadian Institute for Cybersecurity, University of New Brunswick, 2021. [Online]. Available: https://www.unb.ca/cic/datasets/encrypted-dns.html

[26] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 379, no. 2191, pp. 20200202, 2021.doi: [10.1098/rsta.2020.0202]

[27] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 6, no. 1, pp. 1–39, Mar. 2012. doi: [10.1145/2133360.2133363]

[28] A. An and J. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Special Section on Cybersecurity, IEEE Access, vol. 6, pp. 74996–75007, 2018. doi: [10.1109/ACCESS.2018.2882007]

[29] S. Sattar, S. Khan, M. I. Khan, A. Akhmediyarova, O. Mamyrbayev, D. Kassymova, D. Oralbekova & J. Alimkulova, "Anomaly detection in encrypted network traffic using self-supervised learning," Scientific Reports, vol. 15, 2025, Art. no. 26585, doi:10.1038/s41598-025-08568-0.

[30] A. Belenguer, "A Review of Federated Learning Applications in Intrusion Detection," Journal/Publisher, 2025. (or "A federated learning approach to network intrusion detection using residual networks in industrial IoT networks," The Journal of Supercomputing, vol. 80, 2024, pp. 18325–18346, doi:10.1007/s11227-024-06153-2

[31] H. Zhang, J. Ye, W. Huang, X. Liu, and J. Gu, "Survey of federated learning in intrusion detection," Journal of Parallel and Distributed Computing, vol. 195, p. 104976, Jan. 2025, doi: 10.1016/j.jpdc.2024.104976.

[32] W. Marfo, D. K. Tosh, and S. V. Moore, "Network Anomaly Detection Using Federated Learning," MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM), pp. 484–489, Nov. 2022, doi: 10.1109/milcom55135.2022.10017793.

[33] R. Kalakoti, R. Vaarandi, H. Bahşi, and S. Nõmm, "Evaluating Explainable AI for Deep Learning-Based Network Intrusion Detection System Alert Classification," Proceedings of the 11th International Conference on Information Systems Security and Privacy, pp. 47–58, 2025, doi: 10.5220/0013180700003899.

[34] V. Z. Mohale and I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," Frontiers in Artificial Intelligence, vol. 8, Jan. 2025, doi: 10.3389/frai.2025.1526221.