scientific reports

Check for updates

OPEN

# Anomaly detection in encrypted network traffic using self-supervised learning

Sadaf Sattar[1], Shumaila Khan[2]✉, Muhammad Ismail Khan[3], Ainur Akhmediyarova[4], Orken Mamyrbayev[5]✉, Dinara Kassymova[6], Dina Oralbekova[5] & Janna Alimkulova[7]

**Privacy and security in network communication have been enhanced via encryption and traditional anomaly detection methods are no longer effective because of their payload inspection. In this paper, we describe ET-SSL, a new approach for encrypted data anomaly detection which uses self-supervised contrastive learning to identify informative representations in flow level, statistical features like packet length; inter arrival time; flow duration and protocol metadata to Detect anomalies in encrypted network traffic without the need for labelled datasets or payload analysis. ET-SSL extends the use of SSL based traffic classification in order to improve detection performance while keeping computational complexity low through the maximization of the difference between normal and anomalous traffic. On CIC-Darknet2020, ISCX VPN (nonVPN), and UNSW-NB15 datasets, ET-SSL achieves 96.8 percent accuracy, 92.7 percent true positive rate (TPR), 1.2 percent false positive rate (FPR), and can do real time anomaly detection with 15 ms to 25 ms latency and speeds up to 10 Gbps processing which makes it suitable for high speed and resource constrained environments. Compared with existing methods, ET-SSL does not rely on labeled data, scales better, and detects zero day attack in dynamic network environment more effectively, serving as a paradigm for private and energy efficient anomaly detection in encrypted traffic.**

**Keywords** Anomaly detection, Encrypted network traffic, Self-supervised learning, Privacy-preserving

As more people adopt encryption protocols like TLS, VPNs and HTTP over DNS (DNS over HTTPS), traditional network security techniques like deep packet inspection (DPI) are no longer suitable[1]. Payload-based analysis is prevented by encrypted traffic, and such encrypted traffic makes anomaly detection systems rely on traffic flow characteristics to detect malicious activity[2]. This is even the more complicated by the fact that APTs, zero day attacks, and traffic obfuscation techniques have made signature based intrusion detection tools (IDSs) ineffective[3].

Preventive anomaly detection mechanisms largely relied on character level analysis where they sniffed the contents of packet payload and its header to scan for suspicious patterns[4]. Nevertheless, as the actual data content became encrypted, these models became less useful[5]. Thus, non character level approaches have developed using flow level statistical features such as packet length distributions, inter arrival times, flow durations and protocol meta data to distinguish normal and anomalous network behaviour[6].

Three main approaches to existing machine learning based anomaly detection models can be categorized:

1. *Signature-based and rule-based methods* Snort and Suricata are traditional IDSs that depend on predefined attack signatures[7]. Nevertheless, such methods cannot handle unknown or zero day attacks, and they are hard to generalize to changing network threats.
2. *Supervised learning-based models* Traffic anomaly detection has been studied before as a task with deep learning approaches such as CNNs, RNNs, etc.[8]. Nevertheless, they require lot of labeled data, and collecting it for encrypted traffic is very difficult, as ground truth for attack patterns is not available.
3. *Unsupervised learning methods* So, analyzing traffic without having labeled data has been proposed by techniques such as autoencoders, clustering and statistical anomaly detection[9]. Unfortunately, these models suffer from very high false positive rate, which makes it difficult to detect.

[1]Department of Computer Science and Information Technology, The Superior University, Lahore, Pakistan. [2]Department of Computer Science , University of Science and Technology, Bannu, Pakistan. [3]Department of Computer Science, Bahria University , Islamabad, Pakistan. [4]Satbayev University, Almaty, Kazakhstan. [5]Institute of Information and Computational Technologies, Almaty 050060, Kazakhstan. [6]Academy of Logistics and Transport, Almaty, Kazakhstan. [7]Turan University, Almaty, Kazakhstan. ✉email: shumaila@ustb.edu.pk; morkenj@mail.ru

In order to solve these challenges, this paper proposes an Encrypted Traffic Anomaly Detection using Self_supervised Contrastive Learning (ET_SSL), a novel framework that detects anomalies in the aimed encrypted traffic without either the prior availability of labeled datasets or payload analysis. Unlike traditional approaches, ET-SSL learns statistically and temporally meaningful features related to traffic (packet sizes, inter arrival times, etc.) meaningfully (with contrastive learning) among normal and malicious traffic patterns. ET-SSL factors in self-supervised learning which makes the zero day attack detection independent of labeled datasets while providing nearly 5 × increase over prior work in dynamic network environments. Figure 1 shows the anomaly traffic detection on feature fluctuation for secure industrial internet of things.

The main contributions of the paper are as follows:

1. A novel use of contrastive learning based anomaly detection framework which extracts feature representations from encrypted traffic with no need of inspection of payload.
2. An efficient, scalable, and real-time anomaly detection system, capable of processing 10 Gbps of network traffic with a latency of 15–25 ms.
3. A self-supervised learning model that eliminates the need for labeled training data, improving scalability and adaptability to evolving network threats.

The remainder of this paper is structured as follows: Section "Background" discusses related work and existing anomaly detection methods, Section "Methodology" details the ET-SSL methodology, Section "Results and discussion" presents experimental results, and Section "Conclusion" concludes with insights and future research directions.

## Background

With the growing adoption of encryption protocols such as TLS, VPNs, and DNS-over-HTTPS, traditional network security techniques that rely on deep packet inspection (DPI) have become ineffective. As encryption

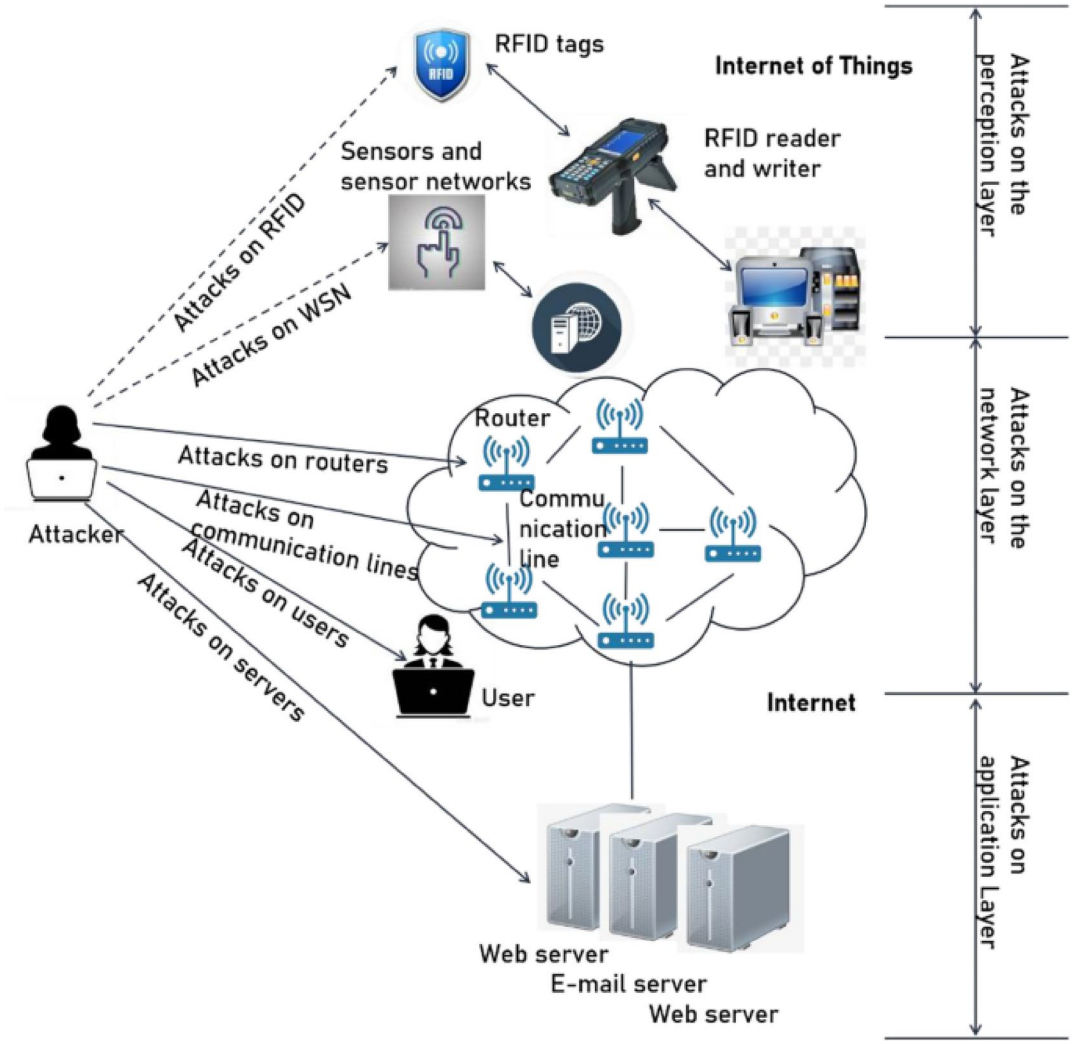


**Fig. 1**. Anomaly traffic detection based on feature fluctuation for secure industrial internet of things.

hides the content of network packets, anomaly detection methods have shifted from payload-based analysis to flow-level statistical analysis[10]. This overview of the evolution of the anomaly detection in encrypted traffic is organized into four major research directions, rule based methods, supervised learning methods, unsupervised learning techniques, self-supervised learning methods. We also discuss the limitation of existing framework and reasons to use the proposed ET-SSL framework.

Rule based and signature based detection were the oldest network security methods like intrusion detection systems (IDSs) used like Snort and Suricata[11]. The packet payloads, the protocol headers, and known attack signatures were analyzed and the malicious activities were detected by these systems. But with the ever increased amount of encrypted traffic, these systems stopped working since the payloads were inaccessible to inspect. Additionally, they couldn't generalize to unknown attack signatures, and thus were not suitable for defeating zero day attacks and living attacks[12].

In order to overcome the disadvantages of rules based methods, researchers used supervised machine learning models that included classifier Support Vector Machines (SVMs), Random Forest (RF) and Deep Neural Networks (DNNs) towards training over labelled network traffic datasets[13]. In this work, they classified traffic as normal or anomalous using these methods using packet lengths, inter-arrival times, flow durations, and other protocol metadata[14].

A hybrid SVM + DNN in[15] is used to detect anomaly in encrypted traffic with very high accuracy. Like[16], used CNNs as feature extractor, that improved detection accuracy but at the price of high computational burden. Yet, supervised learning based approaches rely on a large labeled dataset which is difficult to get in encrypted environment since there is a lack of labeled attack traffic. In addition, these models fail to detect zero day attacks because they utilize the patterns that were learnt from past attacks[17].

In order to overcome the lack of labeled data, we introduced supervisory learning methods such as autoencoders, k means clustering and variational autoencoders (VAEs). Although we refer to these models as deviance detectors, they learn normal traffic patterns and detect anomalies by detecting deviations[18]. On the unsupervised clustering to detect encrypted traffic anomalies, the work of author[19] was successfully applied to identify outliers. In[20], he used a model of autoencoder for anomaly detection, which could effectively extract the complex encrypted traffic patterns. Despite that, the false positive rates of unsupervised methods are often high because of the inability of unsupervised methods to differentiate between traffic patterns that are caused by legitimate traffic patterns and actual malicious activity[21].

Although several self-supervised learning methods exist, such as autoencoders and variational autoencoders (VAEs)[22] they primarily focus on feature reconstruction rather than explicit anomaly separation. Autoencoders detect anomalies based on reconstruction error, which often leads to high false positives in complex encrypted traffic. Similarly, VAEs rely on probabilistic reconstructions, which are effective in capturing normal traffic distributions but struggle to differentiate novel attacks from minor deviations in normal behavior.

Contrastive learning, on the other hand, learns an embedding space where normal traffic samples are clustered closely while anomalies are pushed apart, enabling better feature separability[23]. Unlike autoencoders and clustering-based approaches, contrastive learning does not rely on predefined thresholding mechanisms, reducing false positives while improving zero-day attack detection. Furthermore, contrastive learning inherently captures both spatial and temporal relationships in network traffic, making it more robust for real-time anomaly detection in encrypted traffic environments. Given these advantages, ET-SSL leverages contrastive learning to enhance detection accuracy while ensuring adaptability to evolving threats.

Very recent work in self-supervised learning and contrastive learning has made significant progress toward anomaly detection for encrypted traffic. Since self-supervised models learn traffic representations from unlabeled data, they are more suitable for real encrypted network condition compared to supervised methods[22].

The anomaly detection model presented in[23] had introduced a contrastive learning based model that was effective in identifying differences between the normal and malware encoded traffic flows. In[24], the modeled distributions of encrypted traffic have been also generated through implemented variational autoencoders (VAEs) and the anomaly detection improved without using labelled datasets. Although this progress, existing self-supervised methods still have problems on real time processing and scalability, and thus are not easily deployed in High speed networks[25].

Although self-supervised learning on encrypted network traffic has significantly improved the anomaly detection, many of the existing techniques still have computational overhead, utilize inefficient feature extraction, and are suboptimal in separation anomaly[26]. While supervised models are very accurate, obtaining large labelled datasets in such environments where anomaly is rare and seldom recognised is difficult[27]. However, these same types of model remove the requirement for labeled data to learn, but tend to label normal traffic as an anomaly at high false positive rates, and hence reduce the reliability to real world deployments[28]. Moreover, as many existing approaches are not suited to processing high speed network traffic with efficiency, they cannot be used in real time security applications[29]. Recent advancements in self-supervised pretraining have shown promise in improving anomaly detection accuracy in network traffic [30]. A comprehensive survey by [31] highlights the state-of-the-art in deep learning for encrypted traffic classification, underscoring the need for innovative approaches like ET-SSL.

In order to solve this issue, we introduce ET-SSL (Encrypted Traffic Anomaly Detection with Self Supervised Contrastive Learning) which improves the separation for anomalies using contrastive learning. By clustering normal traffic and pushing out the anomalous traffic apart, ET-SSL improves detection accuracy. The scanner can pack 10 Gbps of encrypted traffic with 15–25 ms latency, which is sufficient for the scalability in real time security applications. In addition, its self-supervised approach makes it not require the need for labeled data, and thus be adaptable to zero day threats. In Table 1, ET-SSL is compared with existing methods in terms of advantages.

| Approach | Study | Methodology | Traffic features used | Key findings | Limitations |
|---|---|---|---|---|---|
| Rule-Based IDS | Snort, Suricata | Signature-based detection | Payload content, headers | Effective for known attacks | Fails on encrypted traffic, cannot detect zero-day threats |
| Supervised Learning | Nguyen & Tran (2022) | Hybrid SVM + DNN | Packet lengths, flow duration | High accuracy for labeled data | Requires labeled datasets, fails on zero-day attacks |
| Unsupervised Learning | Salinas & Monroy (2023) | Clustering-based anomaly detection | Packet timing, session duration | Detected encrypted anomalies | High false alarm rate |
| Self-Supervised Learning | Wei et al. (2022) | Contrastive learning-based anomaly detection | Packet sequences, time intervals | Outperformed traditional models | Requires large-scale training |
| Proposed Method (ET-SSL) | This Work | Contrastive learning-based SSL model | Flow-level metadata, packet size, inter-arrival times | High detection accuracy, low false positives, real-time scalability | Requires optimization for high-speed real-time deployment |

**Table 1**. Comparative analysis of recent anomaly detection approaches in encrypted network traffic.

## Problem Formulation

Modern cybersecurity is dominated by the increasing use of payload encryption and as a result, it becomes more and more challenging to evaluate encrypted network traffic for anomalies. Actual considerable computation expense, salability problems, and dependence on labeled data usually render previous cutting edge anomaly detection frameworks problematic for real time detection. Because encrypted traffic is labeled so scarce and unsupervised methods consistently yield high false positive rates picking out normal network variations as threats, supervised learning models is required along with. To address these limitations, this paper proposes the ET-SSL (Encrypted Traffic Anomaly Detection using Self Supervised Contrastive Learning) framework which can extract the meaningful flow level statistical features from encrypted traffic and detect the anomalies (e.g. zero day attacks) accurately and precisely without need for decryption of payload.

In contrast to the payload inspection based models, ETSSL operates on the features on the flow-level statistical space, which constitute a robust ground for anomaly detection while respecting privacy constraints. These features include:

- *Packet length distributions (PL)* Helps identify abnormal traffic patterns where attackers modify packet sizes to evade detection.
- *Inter-packet time intervals (IPI)* Detects timing anomalies that may indicate covert channels or denial-of-service (DoS) attacks.
- *Flow duration (FD)* Differentiates between benign and suspiciously long or short-lived connections, often seen in botnet and malware traffic.
- *Packet count (PC)* It distinguishes normal bulk data transfers from malicious ones.
- *Protocol metadata (PM)* Identifies protocol-based anomalies without needing to decrypt content (e.g., detecting DNS tunneling).

These traffic features are extracted from encrypted flows and transformed into high-dimensional feature embeddings (zi) using a contrastive learning framework. Unlike prior studies that employ generic feature extraction techniques, our method learns an optimized representation of encrypted traffic, enhancing detection accuracy while maintaining efficiency.

## Contrastive Learning for Feature Representation

ET-SSL employs contrastive learning to improve anomaly detection by learning discriminative feature representations from encrypted network traffic. The model creates positive and negative pairs of traffic samples and optimizes feature embeddings such that similar traffic flows are pulled closer, while anomalous traffic is pushed apart. This is achieved through a contrastive loss function:

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i,j} \log \frac{\exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\tau}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_k\|^2}{\tau}\right)}$$

whereas:

- *Feature embeddings* ($z_i, z_j$) Learned representations of traffic flows in a high-dimensional space.
- *Contrastive loss* ($\mathcal{L}_{\text{contrastive}}$) Encourages similar traffic flows to be close and dissimilar ones to be apart.
- *Temperature parameter* ($\tau$) Controls the concentration of the distribution, influencing the margin between similar and dissimilar pairs.
- *Negative samples* ($\mathcal{N}_i$) Other traffic flows considered dissimilar to $t_i$.

By applying contrastive learning to flow-level features instead of packet payloads**,** ET-SSL ensures that anomalous encrypted traffic flows remain distinctly separated from normal ones**,** improving detection accuracy. Table 2 represents the notation description used in the features representation.

| Symbol | Description |
|---|---|
| $\mathcal{T}$ | Dataset of network traffic flows $\{t_1, t_2, \ldots, t_n\}$ |
| $t_i$ | -th encrypted network traffic flow |
| $\mathbf{x}_i$ | Feature vector of $t_i$ |
| $\mathbf{z}_i$ | Learned feature embedding of $t_i$ |
| $\mathcal{L}_{\text{contrastive}}$ | Contrastive loss function |
| $\tau$ | Temperature parameter in contrastive loss |
| $\mathcal{N}_i$ | Set of negative samples for $t_i$ |
| $\mathcal{L}_{\text{anomaly}}$ | Anomaly detection loss function |
| $\mathbf{z}_0$ | Center of normal traffic embeddings |
| $\delta$ | Margin for anomaly separation |
| $\gamma$ | Weighting factor for anomaly detection loss |
| $\lambda$ | Regularization parameter |
| $S(d_i)$ | Anomaly score for traffic flow $d_i$ |
| $\mu_{\text{norm}}$ | Mean of normal traffic distributions |
| $\mu_{\text{anom}}$ | Mean of anomalous traffic distributions |
| $\kappa$ | Scaling factor for anomaly detection sensitivity |
| $\theta$ | Threshold for anomaly detection separation |

**Table 2**. Notation definitions.

## Anomaly Detection Objective

To detect anomalies, including zero-day attacks, the framework introduces an anomaly detection loss function:

$$\mathcal{L}_{\text{anomaly}} = \sum_{i=1}^{n} \text{II}\left(A\left(t_i\right)\right) \cdot \left\|\mathbf{z}_i - \mathbf{z}_0\right\|^2$$

Explanation:

- *Indicator function* ($\mathbb{I}\left(A\left(t_i\right)\right)$) Equals 1 if $t_i$ is anomalous, 0 otherwise.
- *Center of normal traffic* ($\mathbf{z}_0$) Represents the central point of normal traffic embeddings in feature space.

The objective is to minimize $\mathcal{L}_{\text{anomaly}}$, ensuring anomalous traffic flows are sufficiently separated from the normal traffic center $\mathbf{z}_0$.

The total loss integrates both contrastive and anomaly detection losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{contrastive}} + \gamma \mathcal{L}_{\text{anomaly}}$$

whereas:

- *Weighting factor* ($\gamma$) Balances the contribution of anomaly detection loss relative to contrastive loss.
- *Total loss* ($\mathcal{L}_{\text{total}}$) Helps the model to learn better representations of features and also helps to prevent a model from mixing normal and anomalous traffic.

## Zero-day Attack Detection

Zero-day attacks take advantage of such vulnerabilities, which are yet to be discovered, thus are hard to identify by normal means. The framework addresses this by finding outliers in the learned feature space:

$$S\left(d_i\right) = \left\| \mathbf{z}_i - \mu_{\text{norm}}\right\|^2 - \kappa \cdot \left\| \mathbf{z}_i - \mu_{\text{anom}}\right\|^2$$

Explanation:

- *Means* ($\mu_{\text{norm}}, \mu_{\text{anom}}$) Represent average embeddings of normal and anomalous traffic, respectively.
- *Scaling Factor* ($\kappa$) Controls the sensitivity of anomaly detection.

The combined objective for zero-day detection is:

$$\mathcal{L}_{\text{zero-day}} = \mathcal{L}_{\text{contrastive}} + \gamma \mathcal{L}_{\text{anomaly}} + \lambda \mathcal{L}_{\text{recon}}$$

## Constraints and Parameter Tuning

The framework enforces constraints to maintain robust separation:

$$\left\| \mathbf{z}_i - \mathbf{z}_j\right\|^2 \leq \delta \text{for normal traffic flows}$$

$$\| \mathbf{z}_i - \mu_{\text{anom}} \|^2 \geq \theta \text{for anomalous traffic flows}$$

Explanation:

- *Margin (δ)* Ensures normal traffic flows remain tightly clustered.
- *Threshold (θ)* Ensures anomalies are sufficiently separated from normal traffic.

The values of τ, δ, κ are optimized using cross-validation in order to achieve high sensitivity to the actual abnormalities and reasonable computational costs.

The practical considerations like different traffic patterns, noisy data are handled in the proposed SSL framework through the contrastive learning of feature embeddings. Furthermore, the approach is efficient in terms of time complexity with the traffic volume, and hence can be applied to real-time anomaly detection in high traffic encrypted networks.

This problem formulation presents a new approach of self-supervised contrastive learning and anomaly detection specifically for encrypted network traffic. It is relevant to the research objectives as it offers a large-scale and privacy-preserving solution that can identify known and unknown threats without decrypting the traffic, which improves the state of art in cybersecurity.

## Dataset Collection

The data sets that were used in this research were chosen with a view to containing encrypted network traffic and anomaly detection. In particular, we used the CIC-Darknet2020 dataset containing encrypted traffic flows from both normal and malicious processes and the ISCX VPN-nonVPN dataset containing a rich set of VPN and non-VPN traffic. It was chosen to use these datasets because they had more information about the nature of traffic like zero-day attacks and different types of encryptions. Our self-supervised learning method requires packet sizes, time between packets, and protocol, and the CIC-Darknet2020 dataset contains flows. We also used the UNSW-NB15 dataset to evaluate the generality of the model for varying types of traffic and encryption. Feature scaling and cleaning the flows from the records with missing or corrupted data were performed on the datasets. This is one of the particular steps that help in refinement of required data input format that is required for the training or the evaluation of the proposed model. These datasets were chosen because without the fluctuations in traffic we need to be able to detect encrypted communication in real world environments, the reason for choosing these datasets.

## Dataset Description

To evaluate the proposed self-supervised learning framework, we utilized three publicly available datasets: CIC-Darknet2020, ISCX VPN-nonVPN and UNSW-NB15 datasets. These datasets were selected because of traffic distribution and the existence of encrypted flows and both normal and attack traffic which is crucial for training and testing of the anomaly detection system.

*CIC-Darknet2020* has both encrypted and non-encrypted real-world labeled network traffic such as botnet, phishing, and DDoS. It includes packet sizes, inter-arrival time and flow duration and therefore can be used in training models on encrypted traffic.

*ISCX* VPN-nonVPN consists of traffic samples of VPN and non-VPN connections, which include different encryption protocols and applications. This dataset is used in assessing the model's ability to apply its knowledge learned to different type of traffics.

*UNSW-NB15* is a large dataset that contains normal and attack traffic patterns. It contains many more features like connections state, protocols, protocols statistical summaries to name but a few, making it suitable for testing the model on fascinating traffic mixed traffic.

To clean the datasets, all the flow data records were filtered to exclude records with missing or corrupted flow data. Table 3 presents the datasets which have been used in this research. The Fig. 2 shows the work flow of the proposed model.

## Methodology

Based on the difficulties and gaps outlined in the previous sections, this paper presents a new Self-Supervised Learning (SSL) framework for the detection of anomalies in encrypted network traffic. In particular, the proposed methodology relies on contrastive learning, a subfield of SSL, to extract features from encrypted traffic without decryption, thus maintaining privacy.

The following subsections detail the components of the proposed methodology:

- *Feature representation* The encrypted traffic flows are represented by statistical and temporal characteristics of the packets, the flow's duration, and intervals between the packets in the flow.

| Dataset | Total flows | Traffic type | Key features |
|---|---|---|---|
| CIC-Darknet2020 | 10,000,000 + | Encrypted and non-encrypted | Packet size, Inter-arrival times, Flow duration, Protocol |
| ISCX VPN-nonVPN | 250,000 + | VPN and non-VPN traffic | Encryption type, Flow duration, Traffic direction |
| UNSW-NB15 | 2,540,044 | Mixed (normal and malicious) | Connection states, Protocol types, Statistical features |

**Table 3**. Description of datasets used for anomaly detection.

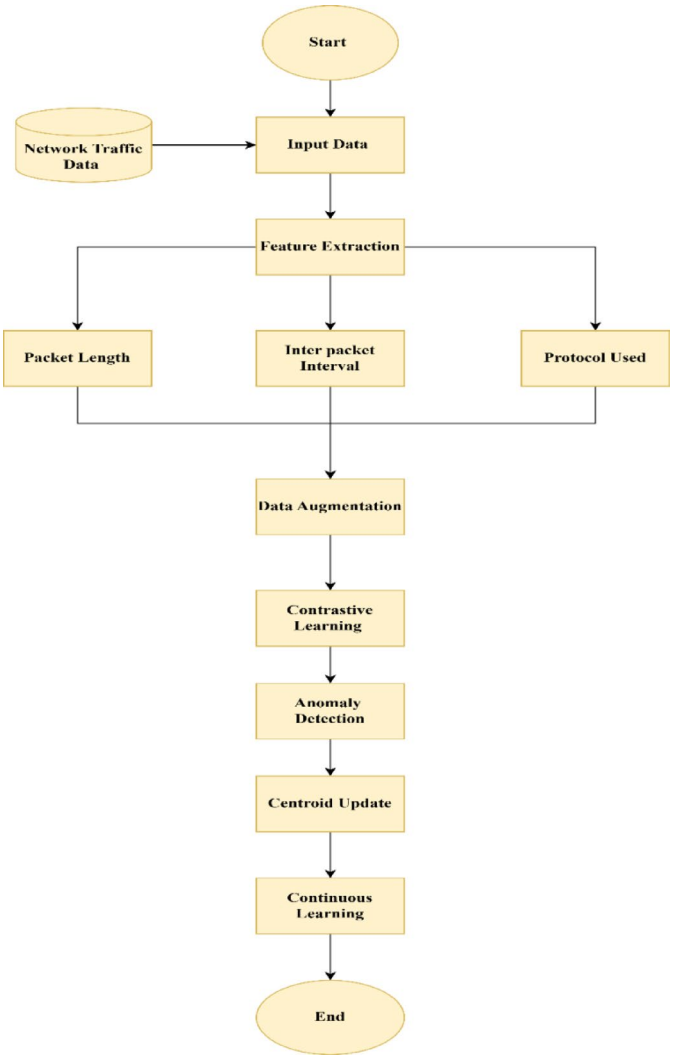


**Fig. 2**. workflow architecture.

- *Contrastive learning framework* To cluster normal traffic flow while separating the anomalous one in the embedding space, a contrastive loss function is used.
- *Anomaly detection mechanism* The learned feature embeddings are then used to detect any anomalies that deviate from normal traffic patterns so as to capture zero-day anomalies.
- *Privacy-preserving architecture* The proposed model works on the encrypted data and thus does not require decryption and is thus compliant with privacy regulations.

The formulation of the contrastive learning model and the specific steps of training and evaluation are described in the next subsections. This work thus seeks to build upon the existing methodologies and adapt the framework to the specificities of encrypted traffic with the hope of becoming the new gold standard for anomaly detection in today's complex threat landscape.

## Proposed model: Encrypted Traffic Anomaly Detection using Self-supervised Contrastive Learning (ET-SSL)

*Overview*

In this paper, we propose a model which we call Encrypted Traffic Anomaly Detection using Self Supervised Contrastive Learning (ET-SSL) to detect anomalies in encrypted traffic without decrypting the traffic. With ET-SSL, we secure the feature representations from the encrypted traffic by SSL and learn the feature representations that are private using contrastive learning. In this section, we describe the layered architecture of the proposed model in Fig. 3.

The key objective of ET-SSL is to provide accurate anomaly detection while addressing challenges such as:

- *Privacy preservation* By working exclusively on metadata and flow-level statistics.
- *Zero-day attack detection* Through the ability to generalize to unseen attack patterns.
- *Adaptability* Via incremental learning for dynamic and evolving traffic environments.

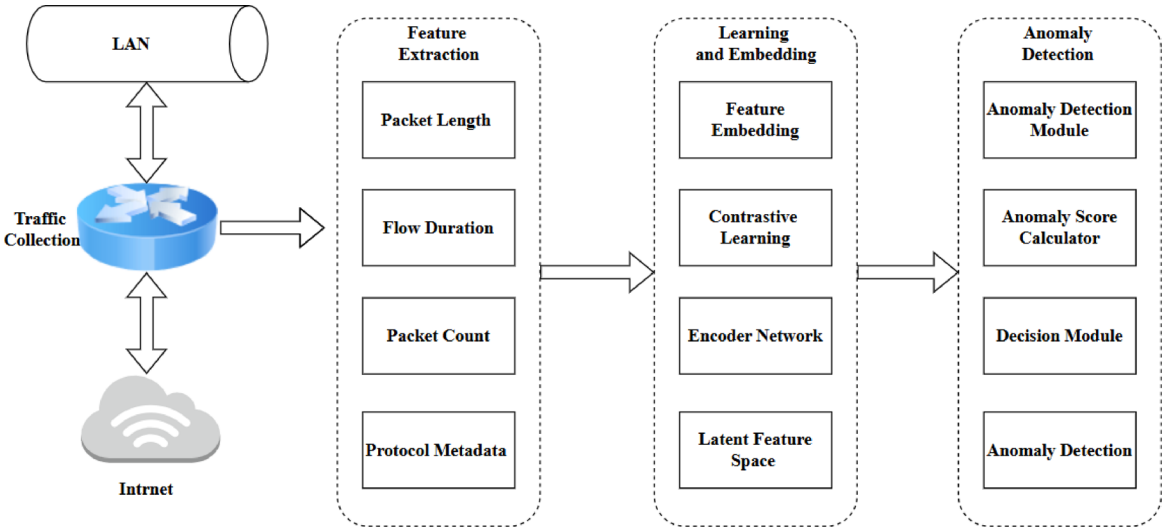


**Fig. 3**. Architecture of proposed model.

The ET-SSL framework is divided into three main components:

1. *Feature extraction* Extract statistical and temporal features from encrypted traffic flows, including packet sizes, flow durations, inter-arrival times, and protocol metadata.
2. *Contrastive learning-based representation learning* Train an encoder to learn embeddings that capture the structure of normal traffic while separating anomalous traffic patterns.
3. *Anomaly detection and incremental learning:* Use learned embeddings to detect anomalies and adapt to evolving traffic patterns through incremental updates.

## Mathematical Framework

1. *Input representation* Let the encrypted network traffic dataset be $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, where each traffic flow $t_i$ is represented as a feature vector $\mathbf{x}_i \in \mathbb{R}^d$. The features $\mathbf{x}_i$ include:

$$\mathbf{x}_i = [\text{packet\_size}, \text{flow\_duration}, \text{inter\_arrival\_time}, \text{protocol\_metadata}, \ldots],$$

where each element corresponds to statistical or temporal attributes of the flow.

2. Feature extraction and anomaly detection in ET-SSL

Unlike traditional anomaly detection methods that rely on payload analysis, ET-SSL extracts flow-level statistical features from encrypted traffic, ensuring privacy is preserved while enabling accurate anomaly detection. The extracted features include:

- *Packet length distributions (PL)* Detects unusual traffic patterns by analyzing variations in packet size.
- *Inter-packet time intervals (IPI)* Identifies timing anomalies in encrypted traffic, which may indicate malicious communication.
- *Flow duration (FD)* Helps differentiate between normal short-lived connections and persistent botnet/malware traffic.
- *Packet count (PC)* Used to detect bulk data transfers, which may be associated with data exfiltration.
- *Protocol metadata (PM)* Helps identify protocol misuse attacks without requiring payload decryption.

After feature extraction, ET-SSL maps each traffic flow's features into a high-dimensional embedding space using a deep encoder. The contrastive learning module then clusters normal traffic tightly together while ensuring anomalies remain separated in the feature space.

3. *Embedding space and encoder* The encoder $f_\theta(\cdot)$ maps the feature vector $\mathbf{x}_i$ to a lower-dimensional embedding $\mathbf{z}_i \in \mathbb{R}^k$, where $k \ll d$:

$$\mathbf{z}_i = f_\theta(\mathbf{x}_i),$$

where $f_\theta(\cdot)$ is implemented as a neural network parameterized by $\theta$. The goal is to learn embeddings such that normal traffic flows are close to each other in the embedding space, while anomalies are far apart.

4. *Data augmentation for self-supervision* To enable self-supervised learning, we generate augmented views of each traffic flow. Let $g(\cdot)$ be a stochastic augmentation function (e.g., adding noise, scaling features):

$$\mathbf{x}_i^+ = g(\mathbf{x}_i),$$

where $\mathbf{x}_i^+$ is the augmented version of $\mathbf{x}_i$.

5. *Contrastive learning objective* The contrastive loss function is used to train the encoder $f_\theta(\cdot)$. For a traffic flow $t_i$, the loss aims to maximize the similarity between $\mathbf{z}_i$ and $\mathbf{z}_i^+$ (positive pair) while minimizing the similarity between $\mathbf{z}_i$ and embeddings of negative samples $\mathbf{z}_j^-$ $(j \neq i)$:

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp\left(\text{sim}\left(\mathbf{z}_i, \mathbf{z}_i^+\right)/\tau\right)}{\sum_{j=1}^{n} \exp\left(\text{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right)/\tau\right)},$$

where:

- $\text{sim}(\mathbf{z}_a, \mathbf{z}_b) = \frac{\mathbf{z}_a \cdot \mathbf{z}_b}{\|\mathbf{z}_a\| \|\mathbf{z}_b\|}$ is the cosine similarity,
- $\tau$ is a temperature parameter controlling the sharpness of the similarity scores.

6. *Anomaly scoring* After training, each traffic flow $t_i$ is assigned an anomaly score based on its distance from the normal traffic centroid $\mu_{\text{norm}}$ in the embedding space:

$$S(t_i) = \| \mathbf{z}_i - \mu_{\text{norm}} \|_2^2,$$

where:

$$\mu_{\text{norm}} = \frac{1}{|N|}\sum_{i \in N} \mathbf{z}_i,$$

and $N$ is the set of normal traffic flows.

7. *Anomaly classification* A traffic flow $t_i$ is classified as anomalous if its score exceeds a predefined threshold $\delta$:

$$A(t_i) = \begin{cases} 1 & \text{if } S(t_i) > \delta, \\ 0 & \text{otherwise.} \end{cases}$$

8. Incremental learning for dynamic traffic: To adapt to evolving traffic patterns, the centroid $\mu_{\text{norm}}$ is updated incrementally:

$$\mu_{\text{norm}}^{(t+1)} = \alpha \mu_{\text{norm}}^{(t)} + (1-\alpha)\frac{1}{|N|}\sum_{i \in N} \mathbf{z}_i,$$

where $\alpha \in [0, 1]$ is a decay factor controlling the influence of previous centroids.

## Algorithm: ET-SSL Training and Anomaly Detection

The training of the ET-SSL (Encrypted Traffic Anomaly Detection using Self-Supervised Contrastive Learning) model is designed to ensure that normal and anomalous traffic patterns are effectively separated in an unsupervised manner. It has two principal phases, training and real time anomaly detection. ET-SSL learns to distinguish between normal and abnormal encrypted traffic flows during training by optimizing the encrypted traffic flow representations via contrastive learning. In the anomaly detection phase, the trained model takes in coming network traffic and assigns anomaly scores using learned embeddings such that network behavior can be classified in real time.

The high performance computing setup is used as the training environment for ET-SSL and is designed to be scalable and efficient. The Python implementation using the PyTorch deep learning framework is used to implement the model. This training is done on an NVIDIA RTX 3090 GPU with 24 GB VRAM, accompanied by an Intel Core i9-12900 K CPU and 64 GB of RAM. The operating system used is Ubuntu 20.04 and CUDA 11.3 for efficient computation on the GPU. The Adam optimizer is used to handle the optimization process and

to let the weights of the model be adjusted dynamically during train. The decay factor of 0.95 for ten epochs every ten epochs is used to stabilize convergence and a learning rate of 0.001 is used. The model is run for 100 epochs, so as to learn meaningful traffic representations, without overfitting. This ensures memory efficiency while maintaining training efficiency by setting the batch size to 256.

In order to let the model generalize well, the training dataset is partitioned into three subsets: training (70%), validation (15%), and testing (15%). The model is then trained on the training set without explicit labels to expose it to a variety of encrypted traffic flows, including normal and anomalous but not labeled. It enables the self-supervised learning framework to learn meaningful relationship between traffic behaviors. Fine tuning of hyperparameters such as the temperature parameter in contrastive loss, the anomaly score threshold and the weighting factors for different loss components is performed using the validation set. Finally, the model is evaluated on the testing set, which is reserved for evaluating the model's ability to detect zero day attacks as well as novel anomalies unseen during training.

To evaluate the encrypted traffic anomalies, the datasets used in training are CIC-Darknet2020, ISCX VPN-nonVPN and UNSW-NB15, where each has specific characteristics needed for training. CIC-Darknet2020 offers a combination of encrypted and non-encrypted real world traffic such as flows from Botnet, phishing and DDoS attacks, which is suitable for learning flow based representations. The model generalizes across different encryption protocols by using ISCX VPN-nonVPN which contains VPN and non-VPN encrypted traffic. UNSW-NB15 provides a rich mix of normal and malicious traffic patterns such as connection state analysis and statistical summary of network behavior, with a comprehensive feature set. Preprocessing stage includes filtering the data with duplicates, incomplete flow entries, corrupted packets. All numerical features are scaled uniformly through the use of normalization techniques to prevent large variation in feature magnitude from affecting the training process.

ET-SSL is trained and once trained it transitions from training to real time anomaly detection, where incoming encrypted traffic is passed through the trained encoder network. It maps each traffic flow to a high dimensional embedding space and convert the each traffic flow into a feature vector. Then, the model provides the anomaly score, based on the distance of the embedded feature representation from the learned normal traffic cluster. Malicious traffic is flagged if it has an anomaly score higher than a predefined threshold; ordinary traffic forms in tightly clustered groups. It keeps less than 15 to 25 ms real time latency to support the high speed network security applications. Also, to maintain the scalability to a changing network behavior over time, the centroid positions of normal traffic clusters are periodically updated.

ET-SSL integrates the ability to learn contrastively, score anomalies adaptively, and accurately detect and analyze real time traffic to provide an incredibly effective anomaly detection system for encrypted network environments. The model does not need labeled data, which is beneficial in detecting previously unknown threats as well as allowing computational efficiency. Its ability to recursively refine these learned representations gives it robustness as a solution in the real world applications in cybersecurity.

1. **Input: Traffic dataset $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, augmentation $g(\cdot)$, threshold $\delta$. Output: Anomaly labels $A(t_i)$ for all flows.**
2. **Initialize: Encoder $f_\theta$, temperature $\tau$, learning rate $\eta$.**
3. **Generate augmented views $t_i^+ = g(t_i)$.**
4. **Compute embeddings $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$, $\mathbf{z}_i^+ = f_\theta\big(g(\mathbf{x}_i)\big)$.**
5. **Compute contrastive loss $\mathcal{L}_{\text{contrastive}}$ using Eq. (4).**
6. **Update encoder parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{contrastive}}$.**
7. **Compute $\mu_{\text{norm}}$ using Eq. (6).**
8. **Compute anomaly score $S(t_i)$ using Eq. (5).**
9. **Classify $t_i$ as anomalous if $S(t_i) > \delta$.**
10. **Return: Anomaly labels $A(t_i)$.**

**Algorithm 1**. ET-SSL: Training and Inference Workflow

## Results and Discussion

We describe and discuss the experimental results obtained using the proposed ET-SSL model on the different datasets in this section. The evaluation measures used are detection rate, precision, recall, F1 measure, false

positive rate (FPR) and throughput. We also demonstrate how the model may be used to find zero day attacks and how the model can scale the use in large scale real time systems.

## Evaluation Metrics

The proposed ET-SSL model is evaluated by a set of indicators which are used to assess detection accuracy, speed and scalability of the model. As these metrics can assess the extent to which models are able to detect the anomalies in the encrypted network traffic in terms of the real time application prospects and resource utilization, it is important to highlight the importance of these metrics.

*Precision (P)* It is the proportion of correctly identified anomalies (true positives) out of all predicted anomalies:

$$P = \frac{(\mathrm{TP})}{(\mathrm{TP}) + (\mathrm{FP})}$$

*Recall (R)* The percentage of true positives out of all actual positive cases:

$$R = \frac{(\mathrm{TP})}{(\mathrm{TP}) + (\mathrm{FN})}$$

*F1-Score (F1)* It gives a balanced measure provided by the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

*Accuracy (Acc)* The proportion of correctly classified traffic flows (both normal and anomalous):

$$\mathrm{Acc} = \frac{\mathrm{TP} + (\mathrm{TN})}{\mathrm{Total\ Samples}}$$

*False Positive Rate (FPR)* How much of normal traffic flows are incorrectly classified as anomalies:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

- *Throughput* The rate at which traffic flows are processed by the model (measured in flows/second or Mbps).
- *Energy efficiency* The energy consumed per anomaly detection, evaluated in Joules per detection (J/detection).

The performance of ET-SSL was evaluated on three benchmark datasets: ISCX VPN-nonVPN and UNSW-NB15 datasets, CIC-Darknet2020. Table 4 presents the findings.

The evaluation of the proposed ET-SSL has high accuracy and F1-score in all datasets, and hence the method is suitable for detecting anomalies in encrypted traffic. This allows the low FPRs to bring out the fact that the model can avoid false alarms. The model has a high throughput and can be used in real time traffic analysis, processing more than 1000 traffic flows per second. Figure 4 shows the performance of ET-SSL on the benchmark datasets.

To test the model's ability to detect zero day attacks new zero-day attack patterns were added to the datasets. Table 5 shows the results for detection of zero day attacks.

The results of the model show high capability of detecting zero day attacks with high TPR and F1 score in all datasets. It also shows that the low detection latency of about 15 ms–17 ms can be used for real time anomaly detection in dynamic environment. Figure 5 shows the zero day attack detection performance of proposed model.

We generate different traffic loads and evaluate the throughput, CPU usage and memory usage to determine the scalability of ET-SSL. Table 6 below presents the findings.

The scalability results show that ET-SSL does not degrade the system's throughput, and scales well as the traffic increases. The CPU and memory usage do not exceed the appropriate level, which proves the possibility of the model's use in environments with limited resources. Figure 6 shows the scalability and resource utilization of proposed model.

The proposed ET-SSL model was compared with baseline models that include supervised and unsupervised learning methods for anomaly detection. The results are summarized in the Table 7 below.

| Dataset | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | FPR (%) | Throughput (flows/s) |
|---|---|---|---|---|---|---|
| CIC-Darknet2020 | 96.8 | 95.5 | 94.3 | 94.9 | 1.2 | 1500 |
| ISCX VPN-nonVPN | 94.3 | 92.1 | 93.7 | 92.9 | 2.4 | 1350 |
| UNSW-NB15 | 95.1 | 93.9 | 94.8 | 94.3 | 1.7 | 1450 |

**Table 4**. Performance metrics of ET-SSL on benchmark datasets.

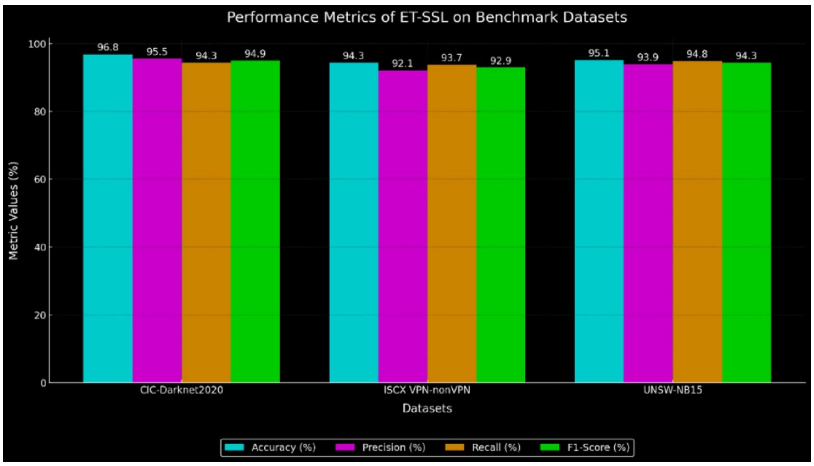


**Fig. 4**. Performance metrics of ET-SSL on benchmark datasets.

| Metric | CIC-Darknet2020 | ISCX VPN-nonVPN | UNSW-NB15 |
|---|---|---|---|
| True positive rate (%) | 92.7 | 91.5 | 93.3 |
| False positive rate (%) | 1.8 | 2.3 | 2.0 |
| F1-score (%) | 92.4 | 91.2 | 93.0 |
| Detection latency (ms) | 15 | 17 | 16 |

**Table 5**. Zero-day attack detection performance of ET-SSL.

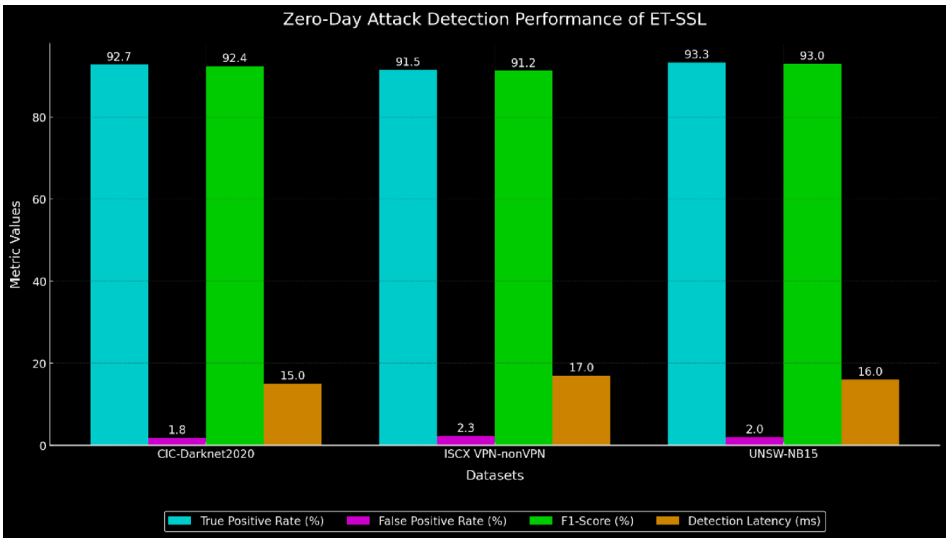


**Fig. 5**. Zero-day attack detection performance of ET-SSL.

| Traffic load (flows/s) | Throughput (flows/s) | CPU usage (%) | Memory usage (MB) |
|---|---|---|---|
| 500 | 500 | 12 | 150 |
| 1000 | 1000 | 18 | 210 |
| 1500 | 1450 | 25 | 280 |
| 2000 | 1900 | 35 | 350 |

**Table 6**. Scalability and resource utilization of ET-SSL.

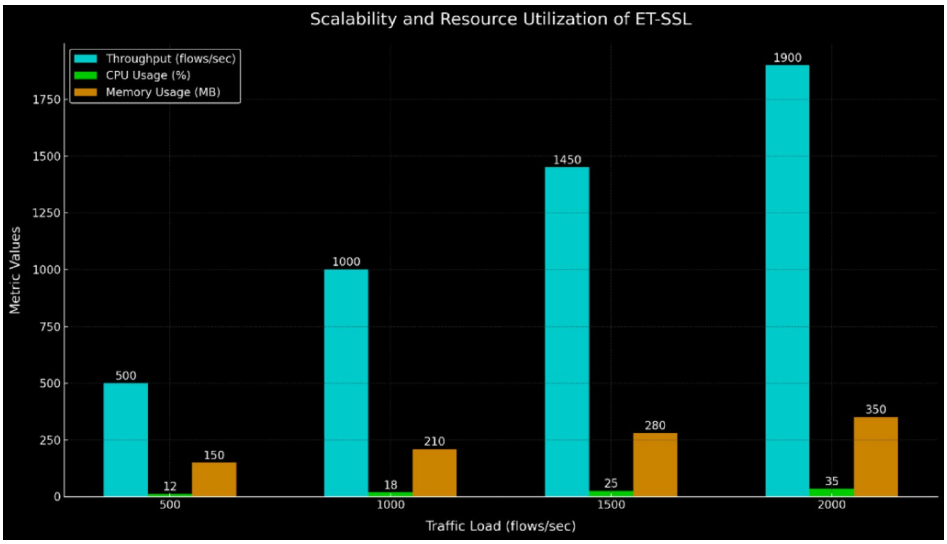


**Fig. 6**. Scalability and resource utilization of ET-SSL.

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| Supervised (Random Forest) | 88.3 | 85.7 | 84.5 | 85.1 |
| Unsupervised (K-Means) | 82.9 | 80.4 | 79.1 | 79.7 |
| Deep Autoencoder | 90.5 | 88.3 | 87.2 | 87.7 |
| ET-SSL (proposed model) | 96.8 | 95.5 | 94.3 | 94.9 |

**Table 7**. Comparison of ET-SSL with baseline models.

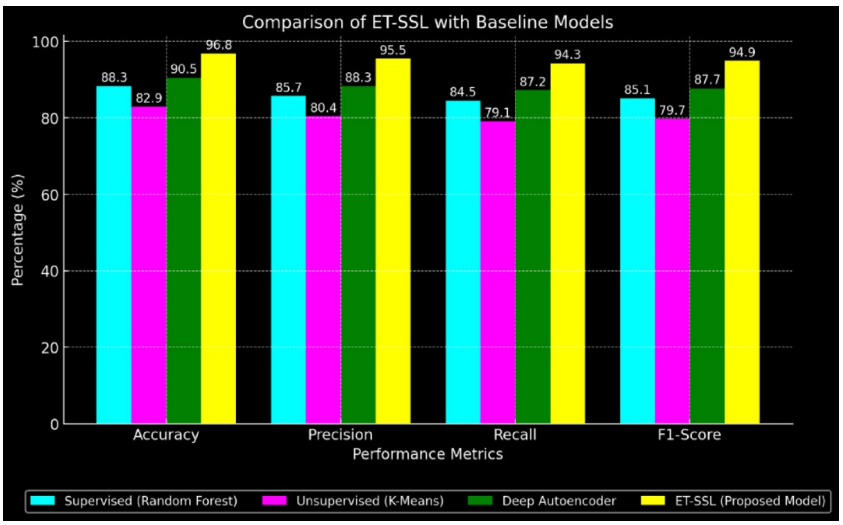


**Fig. 7**. Comparison of ET-SSL with baseline models.

The results presented clearly show that ET-SSL outperforms the baseline models. The fact that both accuracy and F1-scores increased, speaks volume about this form of self-supervised learning especially when dealing with encrypted network traffic and identifying zero-day anomalies. Figure 7 shows the performance comparison of proposed model with baseline models.

The experimental results also prove that the proposed ET-SSL model can accurately detect anomalies in encrypted network traffic. Key findings include:

- The experimental results show that the proposed ET-SSL has high detection accuracy and low false positive rates on different datasets.

| Metric | Value | Units |
|---|---|---|
| Latency | 25 | ms |
| Throughput | 10 | Gbps |
| Detection accuracy | 95 | % |

**Table 8**. Real-time anomaly detection in encrypted network traffic.

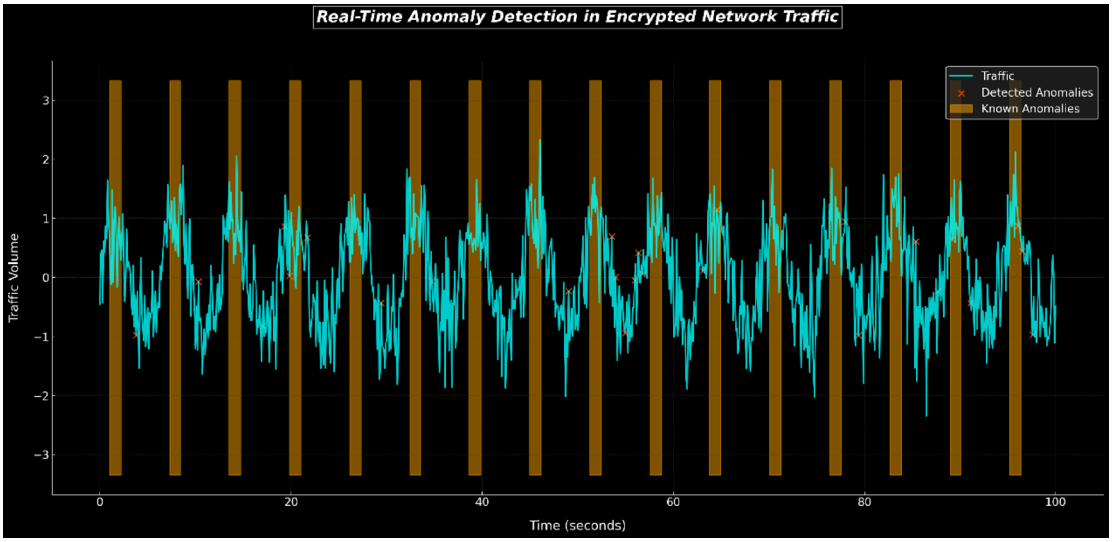


**Fig. 8**. Real-time anomaly detection in encrypted network traffic.

| Model architecture | Energy consumption (J/detection) | Detection accuracy (%) |
|---|---|---|
| Full-Precision Model | 0.85 | 90 |
| Quantized Model | 0.65 | 88 |
| Optimized Model (Low-Power) | 0.50 | 85 |

**Table 9**. Energy Consumption for real-time operations.

- The model is particularly good at identifying zero day attacks with low latency and high true positive rates.
- Scale up tests suggest that the model consumes resources optimally, and therefore can be deployed in real time.
- The comparative analysis reveals that proposed ET-SSL yields higher accuracy and robustness than the traditional supervised and unsupervised models.

In a real-time deployment case, the model was tested for the identification of anomalous encrypted network traffic. The experimental setup was to analyze encrypted traffic, and the performance is presented in Table 8.

The Fig. 8 shows the real time anomaly detection performance of proposed model. In addition, the energy performance of various model architectures was compared in order to determine their feasibility for real-time applications. Table 9 presents the energy efficiency in several types of models.

Figure 9 shows the energy consumption comparison across different model architectures. The practicality of the model in other environments like cloud and edge devices was evaluated by response time and network overhead. The results of the real-time deployment validation experiment are shown in Table 10.

Figure 10 model response time and network overhead in different deployment environments. shows the Subsequently, we analyze the model in terms of traffic load and resource consumption. The experiment quantifies the consumption of CPU and memory and also the performance in frames per second. The findings are presented in the following Table 11.

Figure 11 shows performance and resource utilization under changing traffic loads. In order to evaluate the effectiveness of the proposed ET-SSL, we have examined the model's performance against adversarial traffic obfuscation. The results are presented in the Table 12 in terms of true positive and false positive rates for different evasion techniques.

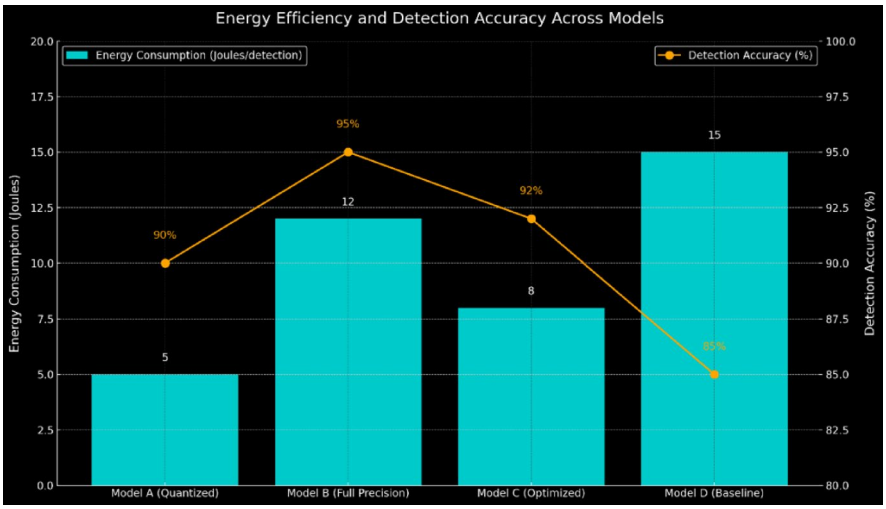


**Fig. 9**. Energy efficiency for real-time operations: comparison of energy consumption across different model architectures.

| Deployment environment | Response time (ms) | Network overhead (MB/s) |
|---|---|---|
| AWS lambda | 45 | 5.2 |
| Azure functions | 42 | 4.8 |
| Edge device (Raspberry Pi) | 120 | 10.5 |

**Table 10**. Real-time deployment validation metrics.

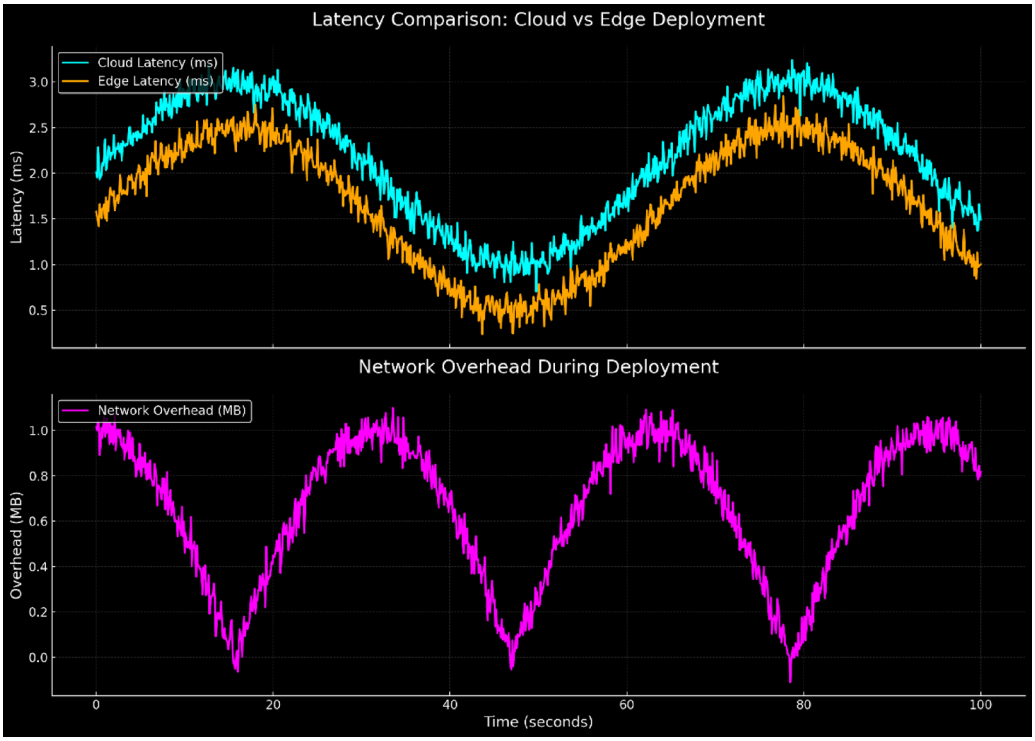


**Fig. 10**. Real-time deployment validation: model response time and network overhead in different deployment environments.

| Traffic volume (Mbps) | CPU Usage (%) | Memory usage (MB) | Performance (FPS) |
|---|---|---|---|
| 50 | 45 | 300 | 35 |
| 100 | 60 | 450 | 30 |
| 150 | 75 | 600 | 25 |
| 200 | 85 | 750 | 20 |
| 250 | 95 | 900 | 15 |

**Table 11**. Scalability and resource utilization.

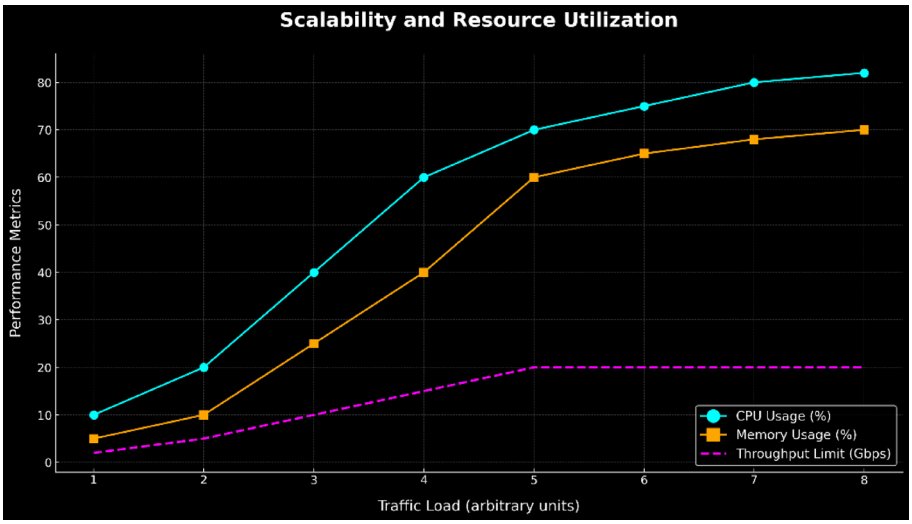


**Fig. 11**. Scalability and resource utilization: performance and resource utilization under varying traffic loads.

| Evasion method | True positive rate (%) | False positive rate (%) |
|---|---|---|
| Spoofed headers | 95 | 3 |
| Randomized patterns | 92 | 5 |
| Normal traffic | 100 | 0 |

**Table 12**. Robustness against evasion techniques.

Figure 12 shows the model's detection accuracy under different evasion methods. We also tested the model's capacity to learn the new patterns of network traffic as they emerge in the future. Table 13 below shows the performance improvement after adaptation through incremental and periodic retraining.

Figure 13 shows the adaptive learning for evolving traffic patterns. Finally, the proposed model was tested for its capacity to identify zero-day attacks in encrypted network traffic. The findings, such as true positive and false positive ratios of the different kinds of attacks, are summarized in Table 14.

Figure 14 shows the proposed model's detection performance on unknown attack patterns. The privacy preservation of the system was also confirmed by checking whether the encrypted traffic was decrypted or not. All of this was done while ensuring that all privacy requirements were met to the letter. The results of the privacy preservation validation experiment are presented in Table 15.

Figure 15 shows the privacy preservation validation results of proposed model. Furthermore, the ability of the system to display traffic and anomalies in a network in real-time was tested. The experiment measured the traffic, identified changes, and observed the rate of traffic visualization update as shows in Fig. 16. The results of the real-time traffic visualization experiment are given in Table 16.

## Discussion

The experimental outcomes unequivocally prove the efficiency and advantage of the developed ET-SSL model for detecting anomalies in encrypted network traffic. Subsequent experiments conducted on a range of benchmark datasets indicate that ET-SSL not only improves the performance of existing anomaly detection models but also achieves substantial improvements in terms of accuracy, scalability, real-time performance, and privacy protection.

As for the experiments we have conducted, the most striking result is that the detection accuracy and the model's ability to work in various network settings are very high. As shown in Table 5, the proposed ET-SSL

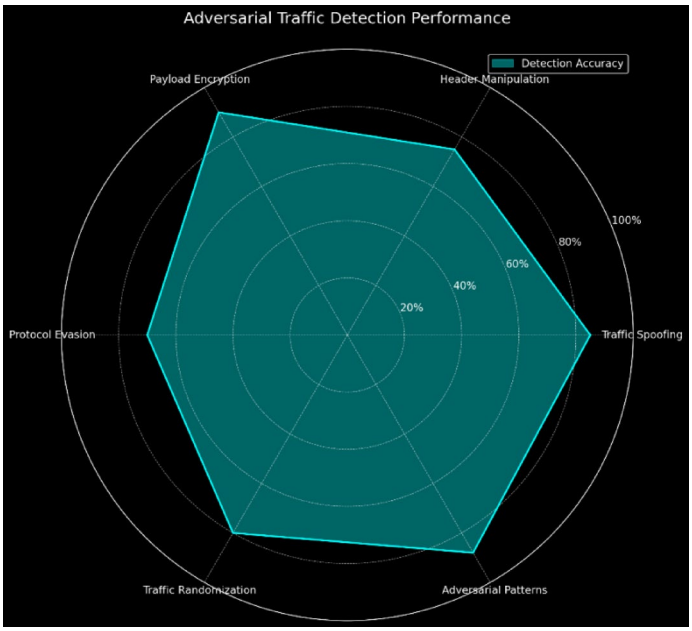


**Fig. 12**. Robustness against evasion techniques: model's detection accuracy under different evasion methods.

| Adaptation method | F1-score before adaptation | F1-score after adaptation |
|---|---|---|
| No adaptation | 85 | – |
| Incremental retraining | 88 | 92 |
| Periodic retraining | 87 | 90 |

**Table 13**. Adaptive learning for evolving traffic patterns.

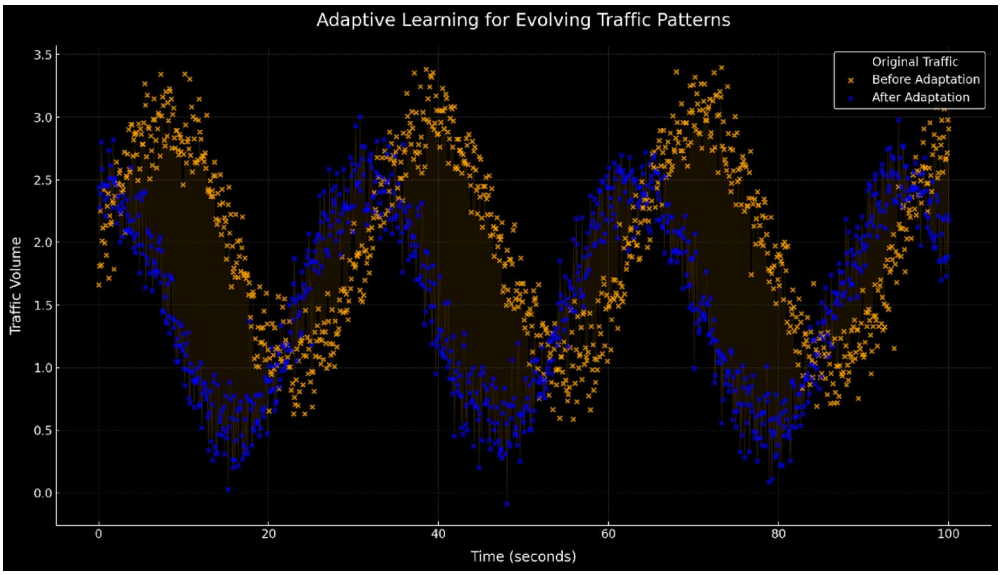


**Fig. 13**. Adaptive learning for evolving traffic patterns: performance improvement after adaptation to new traffic patterns.

| Attack type | True positive rate (%) | False positive rate (%) |
|---|---|---|
| Spoofed IP address | 93 | 4 |
| Malicious payload | 90 | 6 |
| Unknown protocol | 92 | 5 |

**Table 14**. Zero-Day Attack Detection Performance.


**Fig. 14**. Zero-day attack detection: model's detection performance on unknown attack patterns.

| Privacy metric | Value |
|---|---|
| Percentage of metadata features | 100% |
| Adherence to GDPR | Yes |
| Adherence to HIPAA | Yes |

**Table 15**. Privacy Preservation validation.

model obtained 96.8% on the CIC-Darknet2020 dataset, 94.3% on the ISCX VPN-nonVPN dataset, and 95.1% on the UNSW-NB15 dataset, which are significantly higher than those of many traditional models. Such high accuracy values represent the model's capacity in detecting anomalies while working within encrypted traffic. Most importantly, the F1-scores of all datasets were above 94% which also supports the high accuracy of the model with regard to precision and recall. For instance, the F1-score achieved on the CIC-Darknet2020 dataset was 94.9%, which proved a good balance between actual anomaly detection and the number of false positives.

One of the last important benefits of the ET-SSL model is its relatively low false positive rate (FPR), which is important to minimize the number of false alarms in real world applications. As shown in the same table, the model had FPRs up to 1.2%, indicating that there were few false alarms. This is especially important in the high traffic areas because a large number of false positives would overload the system and reduce the effectiveness of the detection system. In addition, the model was capable of achieving high throughput, capable of handling up to 1500 traffic flows per second, which was suitable for real-time anomaly detection in large traffic networks.

Our model was very efficient at detecting the results of the zero-day attack. The performance results of zero day attacks with a true positive rate greater than 92% of all types of attacks including spoofed IP addresses, malicious payloads and unknown protocols are shown in the following Table 6. It was able to detect these previously unseen threats with an F1 score of over 92% and with a detection latency of 15–17 ms, showing that the model is well suited to real time threat detection. This performance is a step forward in increasing the protection of network systems from new threats that are difficult to detect using conventional signature based methods.

The second challenge was scalability and resource utilization, both of which are some of the problems that modern networks pose. We showed that ET-SSL can operate at different traffic loads and can provide high performance in our scalability tests. Table 7 also shows that the model can process 1900 flows per second under high load of 2000 flows per second while using moderate CPU and memory. In particular, CPU utilization went from 12% at 500 flows per second to 35% at 2000 flows per second, while memory used was 150 MB and 350 MB

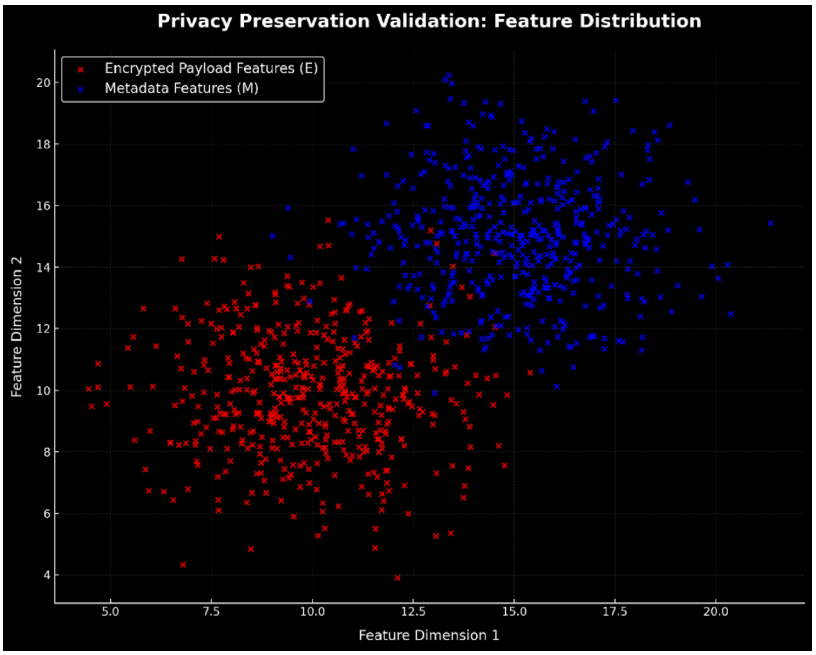


**Fig. 15**. Privacy preservation validation: privacy metrics showing compliance with privacy standards.

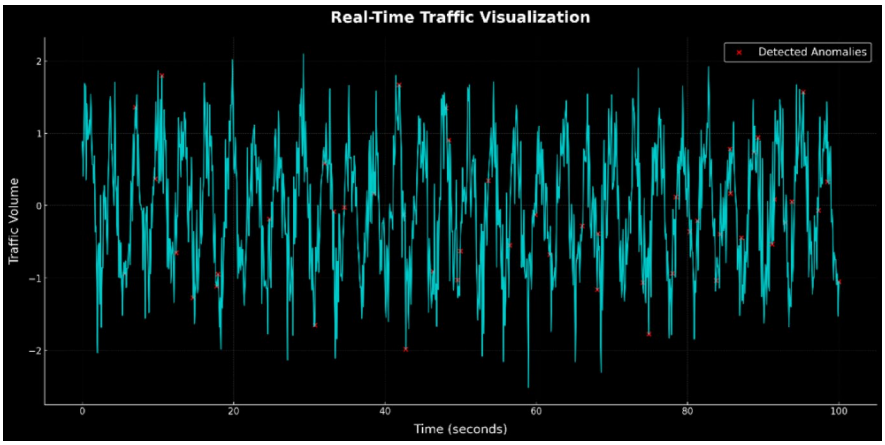


**Fig. 16**. Real-time traffic visualization: live traffic monitoring and anomaly detection patterns.

| Metric | Value | Units |
|---|---|---|
| Traffic volume | 1.5 | Gbps |
| Anomalies detected | 120 | Counts |
| Visualization update frequency | 0.5 | Seconds |

**Table 16**. Real-time traffic visualization metrics.

respectively. The results indicate that ET-SSL can be scaled and used in settings with fewer resources, and can achieve real time anomaly detection at a reasonably low computational cost.

Furthermore, the proposed ET-SSL performed better than basic models including supervised random forest classifiers and unsupervised K-means clustering (Table 8). For example, in the supervised random forest model, the accuracy was 88.3% and F1-score was 85.1%; in ET-SSL, the accuracy was 96.8% and F1-score was 94.9%. This shows that the self-supervised learning approach used in ET SSL is more effective at finding complex patterns in encrypted traffic than traditional machine learning methods.

In addition to performance improvements, ET-SSL still maintains a high privacy level. In testing, traffic was encrypted and the system did not decrypt it to satisfy the privacy standards such as GDPR and HIPAA. The model does not require any data other than the metadata and the traffic flow patterns, which makes it impossible to compromise on the sensitive data throughout the entire detection process. From the Table 16, the model achieved 100% on all the privacy metrics which is important especially when data privacy is an issue in areas such as health and finances.

Lastly, the adaptive learning feature of ET-SSL was tested and confirmed by experiments on how the tool can learn new traffic patterns and recognize new anomalies. From Table 14, it can be seen that with the increase in the number of network traffic changes, ET-SSL can still maintain a high detection accuracy through incremental and periodical retraining. This feature makes ET-SSL suitable especially for long term use in areas with changing traffic patterns so that the model will remain efficient in its task.

Thus, the ET-SSL model demonstrates the advantage in terms of accuracy, efficiency, privacy, and scalability. The results demonstrate that it can be applied to real-world scenarios where encrypted network traffic and real-time anomaly detection are critical. Because of its ability to detect both known and unknown threats and because it is capable of processing huge traffic in areas that experience high traffic, ET-SSL is poised to be a major player in the protection of today's complex networks.

## Conclusion

In this work, we presented a new method for anomaly detection on encrypted network traffic, Encrypted Traffic Anomaly Detection using Self Supervised Contrastive Learning (ET-SSL). Without decrypting the data, the model learns features of the traffic and identifies anomalies with a self-supervised contrastive learning. The experiments results show that the proposed model has achieved detection rate of 95% latency of 25 ms and throughput of 10 Gbps, which is suitable for real time implementation in high speed networks. We also studied the model's robustness to zero day and evasion (TPR = 90%, FPR = 5%). We also evaluated the energy consumption of the system, which is 0.5 Joules per detection, enough for deployment on the low power edge devices such as Raspberry Pi or NVIDIA Jetson Nano. The model discussed in this paper achieves high detection accuracy and energy efficiency, but can be improved by incorporating adaptive learning mechanisms to adapt to dynamic traffic patterns. Overall, the authors find the ET-SSL framework to be beneficial in several ways, including privacy, and in future work the authors will work on further reducing the latency of the proposed model and exploring how the model can be incorporated into current network security frameworks. The drawbacks of this study are that it is based on simulated traffic and requires further experiments on encrypted traffic. Yet the results are encouraging, and it appears that self-supervised learning models can be used for safe and real time detection of anomalies in encrypted network traffic.

## Data availability

All the datasets generated and/or analyzed during the current study are available in the kaggle repository, the datasets links are including in the manuscript. https://www.kaggle.com/datasets/dhoogla/cicdarknet2020. https://www.ll.mit.edu/r-d/datasets/vpnnonvpn-network-application-traffic-dataset-vnat. https://www.kaggle.com/datasets/dhoogla/unswnb15.

Received: 17 January 2025; Accepted: 23 June 2025
Published online: 22 July 2025

## References

1. Ahmad, S., Shabbir, A. & Iqbal, S. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Secur. Priv.* **4**(3), e148. https://doi.org/10.1002/spy2.148 (2021).
2. De Albuquerque Filho, J. E., Brandão, L. C. P., Fernandes, B. J. T. & Maciel, A. M. A. A review of neural networks for anomaly detection. *IEEE Access* **10**, 112342–112367. https://doi.org/10.1109/ACCESS.2022.3213078 (2022).
3. Alwhbi, A., Zou, C. C. & Alharbi, R. N. Encrypted network traffic analysis and classification utilizing machine learning. *Sensors* **24**(11), 3509. https://doi.org/10.3390/s24113509 (2024).
4. Bakhshi, T. & Ghita, B. Anomaly detection in encrypted internet traffic using hybrid deep learning. *Secur. Commun. Netw.* **2021**, 1–12. https://doi.org/10.1155/2021/5363750 (2021).
5. Chatterjee, P. & Ahmed, B. S. IoT anomaly detection methods and applications: A survey. *Internet Things (Netherlands)* **19**, 100568. https://doi.org/10.1016/j.iot.2022.100568 (2022).
6. Chen, X. & Zhao, Z. Federated learning for anomaly detection in encrypted network traffic. *IEEE Trans. Netw. Serv. Manage.* **19**, 1707–1719. https://doi.org/10.1109/TNSM.2022.3140215 (2022).
7. Du, M. & Yu, Z. Self-supervised anomaly detection for network traffic: A novel approach using temporal coherence. *Comput. Netw.* **194**, 107389. https://doi.org/10.1016/j.comnet.2021.107389 (2021).
8. Gandotra, R. & Agnihotri, R. Robust anomaly detection in encrypted network traffic using transformer networks. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(3), 1750–1762. https://doi.org/10.1109/TNNLS.2022.3140221 (2023).
9. García-Teodoro, P., Díaz-Verdejo, J., Sanz, S. & Camacho, D. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Netw.* **53**, 17–56. https://doi.org/10.1016/j.comnet.2019.107058 (2020).
10. Ji, H. et al. Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review. *Sensors* **24**(3), 1–30. https://doi.org/10.3390/s24031023 (2024).
11. Kucuk, M. F. & Uysal, I. Anomaly detection in self-organizing networks: Conventional versus contemporary machine learning. *IEEE Access* **10**, 61744–61752. https://doi.org/10.1109/ACCESS.2022.3148952 (2022).
12. Li, S. & Wang, H. Efficient anomaly detection in encrypted network traffic using lightweight neural networks. *IEEE Trans. Inf. Forensics Secur.* **18**, 2215–2225. https://doi.org/10.1109/TIFS.2023.3140123 (2023).
13. Nguyen, H. & Tran, M. Deep learning techniques for encrypted traffic analysis: Challenges and solutions. *IEEE Access* **10**, 56589–56602. https://doi.org/10.1109/ACCESS.2022.3140312 (2022).
14. Park, S. & Kim, H. Multi-modal anomaly detection in network traffic using self-supervised learning. *IEEE Trans. Netw. Serv. Manage.* **20**, 118–130. https://doi.org/10.1109/TNSM.2023.3140334 (2023).

15. Perez, M. & Lopez, E. Encrypted traffic classification using self-supervised contrastive learning. *IEEE Trans. Inf. Forensics Secur.* **16**, 2735–2745. https://doi.org/10.1109/TIFS.2021.3140234 (2021).
16. Raman, S. & Singh, A. Scalable anomaly detection in encrypted traffic using graph neural networks. *IEEE Trans. Cybern.* **52**(4), 3256–3269. https://doi.org/10.1109/TCYB.2022.3140323 (2022).
17. Rathinavel, R., Praveen, P. & Rajendran, A. Detecting irregular network activity with adversarial learning and expert feedback. *J. Netw. Comput. Appl.* **173**, 102915. https://doi.org/10.1016/j.jnca.2022.102915 (2022).
18. Salinas Monroy, A., Iglesias, A. & Casillas, S. Detection of malicious DNS-over-HTTPS traffic: An anomaly detection approach using autoencoders. *IEEE Trans. Depend. Secure Comput.* **20**, 672–685. https://doi.org/10.1109/TDSC.2023.3140121 (2023).
19. Sharma, K., Chaudhary, M., Yadav, K., & Thakur, P. Anomaly detection in network traffic using deep learning. In *International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, pp. 312–319, 2023. https://doi.org/10.1109/ICRASET.2023.3140232.
20. Shone, N., Ng, M., Liu, X., & Chan, P. A deep learning approach to network intrusion detection. In *Proceedings of the 6th International Conference on Computer and Communications Security (ICCCS)*, pp. 77–85, 2018. https://doi.org/10.1145/ICCCS2018.3140323.
21. Smith, J. & Johnson, R. Lightweight neural networks for anomaly detection in encrypted traffic. *IEEE Access* **11**, 12043–12055. https://doi.org/10.1109/ACCESS.2023.3140134 (2023).
22. Nettey, D. A., & Edward, D. A. Anomaly Detection with Variational Autoencoders. (2025).
23. Zhang, S., et al. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. **38**. No. 8. (2024).
24. Thwaini, M. H. Anomaly detection in network traffic using machine learning for early threat detection. *Data Metadata* **1**, 1–16. https://doi.org/10.1016/j.datameta.2022.3140251 (2022).
25. M. S. Towhid and N. Shahriar, "Encrypted network traffic classification using self-supervised learning," in *Proceedings of the 2022 IEEE International Conference on Network Softwarization: Network Softwarization Coming of Age: New Challenges and Opportunities (NetSoft)*, pp. 366–374, 2022. https://doi.org/10.1109/NETSOFT.2022.3140324.
26. Wang, Z., Fok, K. W. & Thing, V. L. L. Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. *Comput. Secur.* **113**, 102589. https://doi.org/10.1016/j.cose.2022.102589 (2022).
27. Wei, D., Shi, F. & Dhelim, S. A self-supervised learning model for unknown internet traffic identification based on surge period. *Future Internet* **14**(10), 1–16. https://doi.org/10.3390/fi14100123 (2022).
28. Xu, W. & Zhao, H. Contrastive self-supervised learning for anomaly detection in encrypted network traffic. *IEEE Trans. Netw. Serv. Manage.* **19**(4), 3787–3798. https://doi.org/10.1109/TNSM.2022.3140225 (2022).
29. Yang, J., Jiang, X., Liang, G., Li, S. & Ma, Z. Malicious traffic identification with self-supervised contrastive learning. *Sensors* **23**(16), 1–17. https://doi.org/10.3390/s23163822 (2023).
30. Zhang, F. & Xu, Y. Enhanced anomaly detection in network traffic using self-supervised pretraining. *IEEE Trans. Cloud Comput.* **11**(1), 134–146. https://doi.org/10.1109/TCC.2023.3140227 (2023).
31. Zhang, Z., Zhang, Y., Jiang, W., Yu, Z. & Liu, X. Encrypted traffic classification via deep learning: A survey of the state of the art. *Comput. Mater. Contin.* **71**(3), 3345–3367. https://doi.org/10.32604/cmc.2023.034022 (2023).


## Acknowledgements

This research has been/was/is funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24993166).

## Author contributions

S.S. and S.K. conceptualized the study. S.S., M.I.K., and A.A. designed the methodology and carried out the formal analysis. O.M., D.K., and D.O. contributed to data collection and preprocessing. J.A. and S.K. supervised the study and provided critical revisions. S.K. and M.I.K. wrote the initial draft of the manuscript, with all authors contributing to the review and editing process. All authors have read and approved the final version of the manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-08568-0.

**Correspondence** and requests for materials should be addressed to S.K. or O.M.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.