ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 30-42 (2026)
www.astesj.com
Special Issue on Multidisciplinary Frontiers in Engineering, Computing and Applied Sciences

ASTES Journal
ISSN: 2415-6698

TL-SOC: A Hybrid Decision-Centric Intrusion Detection Framework for Security Operations Centers

Imane Lotfi*, Meriem Mandar

Laboratory of Mathematics, Artificial Intelligence and Digital Learning, Higher Normal School of Casablanca, Hassan II University, Casablanca, Morocco

Email(s): imane.lotfi@enscasa.ma (I. Lotfi), m.mandar@enscasa.ma (M. Mandar)

*Corresponding Author: Imane Lotfi, Higher Normal School of Casablanca, Hassan II University, Casablanca, Morocco. Email: imane.lotfi@enscasa.ma

ARTICLE INFO

Article history:
Received: 12 March, 2026
Revised: 27 March, 2026
Accepted: 1 April, 2026
Online: 23 April, 2026

Keywords:
Intrusion Detection Systems
Security Operations Centers
Hybrid Intrusion Detection
Deep Anomaly Detection
Zero-Day Detection
Out-of-Distribution Attacks
Advanced Persistent Threats
Explainable AI

ABSTRACT

Security Operations Centers (SOCs) require intrusion detection systems that achieve high detection accuracy while maintaining a low false-positive rate and robustness to evolving attack patterns. However, most existing machine learning-based approaches primarily focus on detecting known threats and often overlook distribution shifts and the reliability of generated alerts. In this paper, we propose TL-SOC, a decision-centric intrusion detection framework that integrates anomaly representation learning and supervised classification within a unified architecture. The proposed system combines a CNN–Transformer autoencoder, a graph neural network autoencoder, and an XGBoost classifier, whose outputs are fused through a meta-learning decision layer operating under explicit false-positive-rate constraints. To enhance transparency, SHAP-based explanations are incorporated to provide both global and local interpretability of detection outcomes. Experimental results on the CICIDS-2017 dataset show that TL-SOC achieves a precision of 99.63%, a recall of 74.44%, and an F1-score of 85.21%, while maintaining a very low false-positive rate (5.52×10^{-4}). The framework also demonstrates strong robustness under time-series and out-of-distribution scenarios, achieving competitive performance in cross-dataset evaluation. These results highlight the effectiveness of decision-centric hybrid architectures for reliable and operational intrusion detection in SOC environments.

1. Introduction

Lots of research shows how crucial Security Operations Centers (SOCs) are for keeping an eye on networks and finding cyber-attacks as they happen. Within a SOC, Intrusion Detection Systems (IDSs) are really important as they spot anything strange that could be a sign of someone malicious being inside. But today's dangers, for instance zero-day attacks and Advanced Persistent Threats (APTs), are extremely hard to find and predict with the normal machine learning methods of detection; they change and adjust themselves all the time. Right now, most intrusion detection systems (IDS) are overly concerned with correctly naming attacks that are already known, and don't pay enough attention to practical things like adapting to changes in the usual network activity, or making sure they don't give too many false alarms. In fact, Bass and colleagues (as in [1]) said that for cybersecurity systems to be dependable in their judgements, we really need to combine information from multiple sources - and that's much more than just getting the highest possible accuracy. And what's more, recent advances in deep learning have allowed us to find unusual activity, by providing a more complete description of characteristics and a better way to model complex network flows. Naseer and others in [2] showed how well these models work at detecting break-ins. Similarly, in [3] variational autoencoders were used to locate oddities in network traffic and Kwon's detailed review from [4] explained the potential of deep learning for security. However, a single model on its own isn't good at understanding all the different ways an attack might happen and can be unreliable when faced with new or cleverly hidden threats. Because of these drawbacks, many have suggested hybrid or ensemble intrusion detection systems that merge different ways of detecting intrusions, and so offer a more comprehensive approach. In [5], the authors explained a hybrid framework that merges several models. Re-

search [6] builds a hybrid model that combines machine learning and deep learning in the cloud. Also, the reliable detection of multi-model heterogeneous outputs has been studied through decision fusion. In [7], the authors studied methodology of data fusion in intrusion detection. In [8], decision-level fusion improves the real-time detection of cyber-physical systems. Most state-of-the-art techniques are accuracy-centric and fail to tackle the specific challenges that SOCs face, such as the need to control the amount of false positives, the need for robustness to shifts in the input data distribution, and the need for explainable detection. This is particularly important in operational SOC environments, where losing reliable and actionable alerts to the system is a huge problem. This paper presents TL-SOC, a decision-centric hybrid intrusion detection system model that is specifically tailored to the SOC environment. This framework integrates supervised learning for classification and representation learning for anomalies in a unified decision-making architecture that consists of a convolutional neural network and transformer (CNN-Transformer) autoencoder, a graph neural network (GNN) autoencoder, and an XGBoost classifier. These models are integrated in a meta-learner decision layer that operates under stringent false-positive-rate constraints, resulting in highly reliable and accurate detection. An advanced SHAP-based interpretability module is also included to provide explanations at both the feature and decision levels. This consequently assists SOC analysts in their tasks and enhances their confidence in the detection results. This work extends our previous study presented at ICCSC 2025 [9] by introducing the TL-SOC architecture and providing a comprehensive evaluation under realistic conditions, including time-series splits, out-of-distribution scenarios, and cross-dataset validation. These extensions aim to better reflect real-world SOC deployment constraints and assess the robustness and generalization capability of the proposed framework.

1.1. Contributions

This work summarizes several main contributions:

1. **Decision-centric TL-SOC architecture:** We illustrate a hybrid intrusion detection model designed for SOC environments that merges anomaly representation learning with a decision pipeline through supervised learning. The architecture employs a CNN-Transformer autoencoder, a graph neural network autoencoder, and an XGBoost classifier. It is purposefully built to meet operational requirements involving low false positives and dependable alerting.

2. **Meta-learning-based fusion:** We propose, for the first time, a decision-level fusion strategy using meta-learning, capable of dynamically fusing detection signals of varying nature (like anomaly scores and classification probabilities). This allows the model to fully utilize the complementary aspects of the inputs, particularly enhancing the model's performance under distribution shifts and previously unencountered attack patterns.

3. **False-positive-aware calibration:** We introduce a threshold calibration approach that aims to meet specific constraints on the false-positive rate. This calibration aligns the detection process with the operational requirements of the SOC by reducing alert fatigue while sustaining a high level of detection.

4. **Advanced explainability:** We offer participants the opportunity to employ a SHAP-based framework for the purposes of interpretability. This approach provides global feature attribution and local decision explanations, which together improve the opacity of the process and assist SOC analysts in comprehension and sanctioning the detection results.

5. **Robustness evaluation:** Assessing the model's robustness is done via a thorough evaluation under realistic settings. Evaluations include time-series splits to avoid data leakage, zero-day attack simulations, out-of-distribution scenarios (APT-like attacks), cross-dataset tests. This set of experiments seeks to determine the generalization capabilities of the proposed framework.

6. **Comparative analysis:** We have carried out comprehensive testing for numerous intrusion detection systems, including anomaly-based models, supervised models, and hybrid models. The tradeoff between detection performance and robustness illustrates the value of the TL-SOC framework when operational constraints are imposed.

2. Related Work

2.1. Machine Learning for Intrusion Detection

Techniques used in Artificial Intelligence get employed in Network Intrusion Detection Systems as they can detect anomalous patterns in a network and traffic pattern recordings. In extensive Networks, Deep Neural Networks are proficient in identifying malicious activity.

In [2], the authors explain how deep learning can result in a higher level of detection for network traffic, thanks to the learning of hierarchies. One of the most popular techniques, and the focus of much recent research, is the autoencoder and its application to anomaly detection in network traffic by learning a compact representation of normal traffic and detecting anomalies based on reconstruction error. Similar to [10], the authors of the current work propose an autoencoder-based approach to network anomaly detection. Similarly, in [3] a variational autoencoder was proposed to enhance the detection of anomalies in network flow data. In [11], an LSTM-based autoencoder was used to capture the temporal aspects of network traffic. The authors in [4] present an excellent and comprehensive survey on deep learning-based techniques for the detection of network anomalies. All these advancements notwithstanding, when applied in practice, standalone deep learning solutions still face significant challenges. In particular, these models are typically limited in terms of adaptable and reliable decision thresholds, as well as interpretability, and they tend to lose their robustness to novel attack patterns. This fosters the need for hybrid and decision-centric approaches that are more in tune with the actual requirements of a SOC.

2.2. Hybrid and Ensemble Intrusion Detection Systems

To enhance performance and increase the reliability of detection systems, a number of approaches have proposed hybrid intrusion detection systems based on the integration of different machine learning and deep learning techniques.

In [5], the authors suggested a hybrid architecture consisting of deep learning, random forest, and clustering models, with the aim of improving detection performance. Also, [6] proposed a hybrid model that combined machine learning and deep learning for intrusion detection in cloud environments. Other research work has focused on the use of ensemble and stacking techniques for performance improvement. Authors of [12] proposed a model of stacked ensemble learning for wireless network intrusion detection. [13] developed a stacking ensemble of deep learning models in order to detect intrusions in the Internet of Things. Furthermore, in [14] the authors proposed HDLNIDS, which is described as a hybrid deep learning based intrusion detection system that combines various deep learning components. However, in spite of these innovations, the majority of hybrid and ensemble approaches focus primarily on boosting classification performance under in-distribution conditions. Likewise, they often ignore operational constraints, such as control of the false positive rate, decision tuning, and robustness regarding distributional shifts. This limits the approaches to real world SOC applications, where detection systems must operate under the conditions of constant change and novel attack scenarios. This gap aims for the development of decision-centric frameworks, including the proposed TL-SOC, which integrate heterogeneous detection mechanisms in an operationally bound context.

2.3. Decision Fusion and Meta-Learning in Cybersecurity

Decision fusion methods have been refined to improve the robustness and reliability of intrusion detection systems.

The initial work in [1] applied the principles of multisensory data fusion to cybersecurity decision-making. More recent work has examined data-level and feature-level fusion approaches for synthesizing multiple detection models. While an example of feature fusion was reported to improve the performance of deep learning-based intrusion detection systems in [15], the authors of [7]examined data fusion methods for intrusion detection. Methods for decision fusion were used to consolidate the outputs of different models in a collaborative manner. In [8], the authors illustrate the usefulness of decision fusion for real-time intrusion detection with higher detection reliability in industrial cyber-physical systems. More recent works have used meta-learning as a candidate for flexible decision-making in intrusion detection. The author of [16] proposes a meta-learning structure for intrusion detection with limited attempts. Likewise, in [17] show that such techniques can enhance the detection of anomalies in cyber-physical systems. These techniques allow models to combine multiple data sources and adapt to novel attack strategies. Despite significant progress in the field, most of the works focus on enhancing the detection capability and overlook the equally important operational constraints, such as the need to maintain a low false alarm rate, the loss of decision reliability, and robustness to changes in the probability distribution. These shortcomings are likely to affect the operational applicability of systems located in a security operations center (SOC), where intrusion detection systems are integrated in a highly reliable and stable manner.

To mitigate these issues, this research proposes **TL-SOC**. This is a decision-centric intrusion detection framework that employs deep learning for anomaly representations, hybrid detection models, and decision fusion via meta-learning, all constrained by a defined false positive rate. This approach enables robust, interpretable, and well-calibrated detection, which is particularly beneficial for SOCs.

3. Proposed TL-SOC Framework

3.1. Overview of the TL-SOC Architecture

The innovative TL-SOC architecture aims to improve cyberattack detection and decision support in security operations centers (SOCs) through an integrated pipeline of deep anomaly modeling, supervised validation, and explainable alerts. Figure 1 illustrates this pipeline. An autoencoder, designed using CNN and Transformer architectures, learns the latent representations of normal traffic and those attributable to reconstruction error, then generates anomaly scores. These scores are then refined by a supervised post-filtering model based on XGBoost, which improves the discrimination of activity (benign or malicious) and thus reduces the number of false positives. Alerts are triggered by a post-filtering model based on a calibrated threshold to limit the false positive rate. The system also incorporates a SHAP explainability model to provide SOC analysts with a detailed explanation of the alert, identifying the most critical features. Consequently, the architecture promises reasonably reliable detection of known attacks and emerging risks such as advanced persistent threats (APTs) and zero-day attacks.

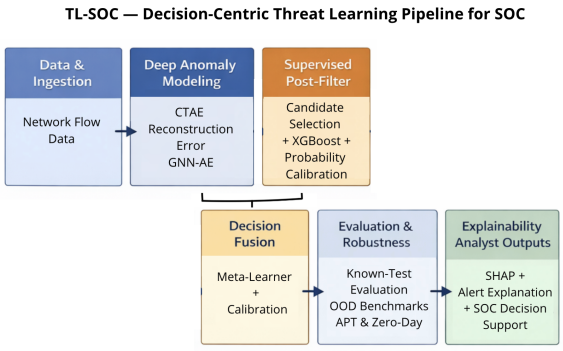

Figure 1: Overview of the proposed TL-SOC pipeline combining deep anomaly modeling, supervised decision refinement, and explainable SOC alert analysis.

3.2. Representation Learning with CNN–Transformer Autoencoder

The first step in the TL-SOC pipeline consists of anomaly-based representation learning. This learning process, which is part of the TL-SOC pipeline, uses a CNN-Transformer autoencoder (CTAE) to learn representations applied to tabular network flow data. After preprocessing steps (cleaning, encoding, and normalization), each network flow is represented by a feature vector $\mathbf{x} \in \mathbb{R}^d$. With the development of end-to-end models, an increasing number of hybrid models integrating different components

are emerging. Examples include convolutional neural networks (CNNs) and Transformer-type architectures. CNNs excel at handling local interactions, while Transformers extend to global interactions through self-attention. Composed of multiple layers of each type of module, fully trained layers should be capable of capturing dependencies at both the local and global levels within their respective domains.

$$\mathbf{z} = f_\theta(\mathbf{x}), \tag{1}$$

where f_θ denotes the encoder parameters. The decoder then reconstructs the original feature vector

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z}), \tag{2}$$

where g_ϕ represents the decoder function.

The autoencoder was able to learn a compact representation of typical network behavior because the model was trained only on typical traffic. The reconstruction error between the input and its reconstruction is used to identify anomalies during inference. The anomaly score produced by the CTAE module is defined as

$$s_{\mathrm{AE}}(\mathbf{x}) = \frac{1}{d}\,\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\,, \tag{3}$$

where d denotes the number of input features. Higher reconstruction errors indicate stronger deviations from learned normal patterns and thus correspond to higher anomaly likelihood. The resulting score sAE(x) constitutes the first signal used by the TL-SOC fusion layer described later in Section 3.5.

3.3. Graph Neural Network Autoencoder

Neural networks (NNs) are used in many artificial intelligence (AI) applications. Even in the design and development of traffic monitoring systems, certain modern methods are being implemented. The TL-SOC framework uses a generalized neural network-based autoencoder (GNN-AE). Unlike traditional autoencoders, where each instance is processed independently, the GNN-AE encodes a graphical model of the traffic space to capture local dependencies.

Let $x_i \in \mathbb{R}^d$ denote the feature vector representing the i-th network flow after preprocessing and normalization. A graph $G = (V, E)$ is presented where each node $v_i \in V$ corresponds to a flow sample x_i. The edges E are defined using a k-nearest neighbor (k-NN) strategy based on feature similarity. For each node v_i, a set of neighbors $\mathcal{N}(i)$ is obtained by selecting the k closest samples in the feature space using the Euclidean distance:

$$\mathcal{N}(i) = \arg\operatorname{topk}_{j\neq i}\ \|x_i - x_j\|_2 \tag{4}$$

The local graph is created to illustrate the adjacency structure. In this graph, network flows are closely linked. To design it, a small sample of normal traffic is used to describe the distribution and topology of typical traffic. For each node v_i, the information from its neighborhood is aggregated using a mean aggregation operator:

$$h_i = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} x_j \tag{5}$$

The pair (x_i, h_i) forms the input to the GNN autoencoder. The encoder maps these inputs into a latent representation z_i through a nonlinear transformation:

$$z_i = \sigma\left(W_e[x_i \,\|\, h_i] + b_e\right) \tag{6}$$

where $[\cdot\|\cdot]$ denotes feature concatenation, W_e and b_e are learnable parameters, and $\sigma(\cdot)$ is a nonlinear activation function. The decoder then reconstructs the original flow representation:

$$\hat{x}_i = W_d z_i + b_d \tag{7}$$

The model is trained to minimize the reconstruction loss over benign samples:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|_2^2 \tag{8}$$

During inference, flows that deviate significantly from the learned structure produce larger reconstruction errors. The anomaly score produced by the GNN-AE is therefore defined as

$$s_{gnn}(x_i) = \|x_i - \hat{x}_i\|_2^2 \tag{9}$$

This graph-based anomaly representation complements the reconstruction signal produced by the CNN–Transformer autoencoder. While the CTAE captures feature-level dependencies within traffic flows, the GNN-AE models relational similarities between flows, enabling the TL-SOC framework to detect anomalies that arise from structural deviations in the traffic space.

3.4. Supervised Post-Filtering with XGBoost

To finalize the detection process, TL-SOC integrates a supervised classification module based on XGBoost. While autoencoder-based models identify deviations from normal behavior, supervised learning makes it possible to capture the discriminating patterns associated with known malicious activities.

Let $\mathbf{x} \in \mathbb{R}^d$ be the normalized feature vector representing a network flow. The XGBoost classifier learns a function that estimates the probability that the flow corresponds to malicious activity. Formally, the predicted probability is expressed as

$$p_{\mathrm{XGB}}(\mathbf{x}) = \sigma\left(\sum_{k=1}^{K} f_k(\mathbf{x})\right), \tag{10}$$

where f_k denotes the k-th regression tree in the ensemble, K is the number of trees, and $\sigma(\cdot)$ is the logistic function transforming the aggregated tree outputs into a probability score. The resulting probability $p_{\mathrm{XGB}}(\mathbf{x})$ represents the supervised detection signal of the TL-SOC pipeline. This score is then combined with the anomaly scores produced by the CTAE and GNN modules within the meta-fusion layer described below..

3.5. Decision Fusion and Threshold Calibration

To produce a reliable detection decision, the TL-SOC framework combines the outputs of the anomaly detection and supervised classification modules through a meta-level fusion step that integrates complementary signals from the CNN-Transformer autoencoder, the GNN-based anomaly model, and the XGBoost classifier.

Let $s_{\mathrm{AE}}(\mathbf{x})$ be the anomaly score generated by the CTAE module, $s_{\mathrm{GNN}}(\mathbf{x})$ be the anomaly score produced by the GNN-based model, and $p_{\mathrm{XGB}}(\mathbf{x})$ be the probability of malicious intent estimated by XGBoost. These signals are first aggregated into a meta-feature vector.

$$\mathbf{m}(\mathbf{x}) = [s_{\mathrm{AE}}(\mathbf{x}), s_{\mathrm{GNN}}(\mathbf{x}), p_{\mathrm{XGB}}(\mathbf{x})]. \tag{11}$$

A logistic regression meta-learner then combines these features to produce an intermediate detection score

$$p_{\mathrm{raw}}(\mathbf{x}) = \sigma\Big(\mathbf{w}^{\top}\mathbf{m}(\mathbf{x}) + b\Big), \tag{12}$$

where $\mathbf{w}$ denotes the learned fusion weights, b is a bias term, and $\sigma(\cdot)$ is the logistic function. To improve probability reliability, the raw score is further calibrated using a monotonic calibration function $C(\cdot)$ (implemented through isotonic regression in our pipeline):

$$F(\mathbf{x}) = C(p_{\mathrm{raw}}(\mathbf{x}))\,. \tag{13}$$

Finally, the alert decision is obtained by applying a threshold selected under a false-positive-rate constraint:

$$y(\mathbf{x}) = \mathbb{I}(F(\mathbf{x}) \geq \tau_{\mathrm{FPR}})\,, \tag{14}$$

where τ_{FPR} is chosen on validation data to satisfy a target false positive rate. This decision-centric formulation ensures stable alert generation in SOC environments while maintaining strong detection capability.

3.6. Explainable Alert Analysis (SHAP)

For the purpose of building trust with customers through transparency, the TL-SOC framework incorporates an explainability module based on SHAP (SHapley Additive ExPlanations). This module provides explainability at the level of features for pipeline alerts, making it possible for SOC analysts to gain insight into the reasons for the abnormal behavior. Let $F(\mathbf{x})$ be the calibrated detection score produced by the TL-SOC pipeline for a network flow $\mathbf{x}$. SHAP explains the prediction by assigning a contribution value ϕ_j to each input feature x_j. The model output can therefore be expressed as

$$F(\mathbf{x}) = \phi_0 + \sum_{j=1}^{d} \phi_j, \tag{15}$$

where ϕ_0 represents the baseline prediction and ϕ_j denotes the Shapley contribution of feature j.

These contributions quantify how each feature influences the final detection score. Positive values present that the feature increases the likelihood of an anomaly, while negative values reduce it. By highlighting the most influential features for each alert, SHAP provides transparent explanations that facilitate rapid investigation and decision-making by SOC analysts.

3.7. Design Rationale

Balancing the need for different, mutually beneficial detection approaches to overcome the issues associated with single-model intrusion detection systems in SOC environments is the inspiration behind the TL-SOC design. The CTAE has been used to capture both local feature interactions and global dependencies in the context of network traffic data. Convolutional layers have been used to learn localized patterns, while the transformer encoder is used to capture relationships over long distances, thus facilitating effective learning of anomalies. The GNN-AE has been proposed to learn the structural relationships of network flows. By using the k-nearest neighbor graph approach, GNN-AE is able to learn the similarities among the traffic instances and to detect anomalies concerning the underlying data structure. The XGBoost classifier, a supervised learning module, has been proposed to enhance the identification of benign and malicious traffic. Its potential to model intricate decision boundaries and to provide a ranking of feature significance renders it especially appropriate for enhancing the detection of anomalies. Finally, a meta-learning decision layer is used to integrate these different signals. This fusion mechanism adjusts the weights of the integrated information sources to enhance robustness to distribution shifts and to make detection decisions more reliable when the threshold for false positives is tight.

4. Experimental Setup

4.1. Datasets

The experiments were conducted using two publicly available reference datasets: CICIDS-2017 and CSE-CIC-IDS2018, both widely used in network intrusion detection research.

The CICIDS-2017 dataset contains realistic network traffic, including both benign activities and multiple attack scenarios such as brute-force attacks, denial-of-service (DoS), distributed denial-of-service (DDoS), infiltration, botnet activity, and web attacks. Also, each network flow is represented by a set of statistical characteristics extracted from packet-level information. After preprocessing and feature selection, each flow is represented by a numeric vector $\mathbf{x} \in \mathbb{R}^d$.

Table 1: Summary of the CICIDS-2017 dataset used in the experiments.

Property	Value
Dataset	CICIDS-2017
Total flows	2.57M
Benign flows	2.14M
Attack flows	0.42M
Attack categories	DoS, DDoS, Brute Force, Web Attacks, Botnet, Infiltration
Features used	70

The CSE-CIC-IDS2018 dataset is used for cross-dataset evaluation to assess the generalization capability of the proposed framework under distribution shift. It contains several types of attacks and more diverse traffic patterns, making it suitable for assessing robustness in heterogeneous network environments.

Table 2: Summary of the CSE-CIC-IDS2018 dataset used for cross-dataset evaluation.

Property	Value
Dataset	CSE-CIC-IDS2018
Total flows	1,869,101
Benign flows	1,660,687
Attack flows	208,414
Attack categories	DoS attacks-GoldenEye, DoS attacks-Slowloris, FTP-BruteForce, SSH-Bruteforce
Features used	70
Usage	Cross-dataset evaluation (test only)

The dataset is partitioned using both stratified and chronological splitting strategies. The stratified split is used for baseline evaluation, while the time-series split is adopted to simulate realistic SOC conditions and prevent data leakage. For cross-dataset evaluation, the model is trained on CICIDS-2017 and tested on CSE-CIC-IDS2018 without retraining.

4.2. Data Preprocessing

Before training them, several preprocessing steps were applied to prepare the network flow data.

Let $\mathbf{x} = (x_1, \ldots, x_d)$ be the feature vector representing a network flow. First, corrupted or incomplete records were removed from the dataset. Next, the categorical features were transformed and converted into numerical representations using encoding techniques such as label encoding or one-hot encoding. All numerical features were then normalized by min-max scaling to ensure comparable value ranges between different features. For each feature x_j, the normalization is defined as

$$x'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{16}$$

where $\min(x_j)$ and $\max(x_j)$ denote the minimum and maximum values of feature j observed in the training data.

After normalization, each feature value is projected into the interval $[0, 1]$, producing the normalized feature vector $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_d)$. To prevent data leakage, the normalization parameters $\min(x_j)$ and $\max(x_j)$ were estimated using only the training dataset and then applied unchanged to the validation and test sets.

4.3. Model Training

The dataset was divided into training, validation, and test portions using stratified sampling. All training set models for anomaly detection were trained on benign traffic to learn the benign normal behavior of network flows. Validation set was used for calibration and thresholding and the test set was used for final evaluation. Before training begins, numerical dataset features were processed via imputation of missing values, quantile clipping using benign training samples, and MinMax normalization which was fit to normal data. The Adam optimizer was employed to train the CTAE model, with early stopping based on validation loss. Reconstruction errors were turned into anomaly scores. When the GNN-AE was enabled, it learns local relationships using a k-nearest neighbors graph built from benign samples. Additionally, a supervised post-filtering module based on XGBoost was trained on attack samples and a few hard normal samples from the validation set. The trained post-filtering was calibrated using isotonic regression. A logistic regression meta-learner combined the outputs of all these modules, and the false positive rate was set to determine the final threshold on the validation set.

4.4. Implementation Details

Using a workstation with a multi-core processor and ample memory for large-scale network traffic data processing, all experiments leveraged standard machine learning and deep learning libraries such as TensorFlow/Keras, PyTorch, and XGBoost, implemented in Python.

Tabular network flow features are processed using the CTAE architecture. Convolutional layers and multi-head self-attention are features of the Transformer encoder blocks. The integration dimension was set to 64, with 4 attention heads and a forward dimension of 128. The model was developed using the Adam optimizer, with a learning rate of 10^{-3}, a batch size of 1024, and early stopping with respect to validation loss. For the KNN graph, the GNN-AE uses $k = 10$ with respect to Euclidean distance between normalized feature vectors. A simple aggregation mechanism minimizes the reconstruction loss on benign samples. The XGBoost classifier was set to 300 estimators, a max depth of 6, a learning rate of 0.05, and subsampling at 0.8 for both instances and features. Imbalance in classes was corrected with the use of a scale_pos_weight parameter derived from the training data. Using logistic regression, the meta-learning fusion layer combines anomaly scores from CTAE and GNN-AE with XGBoost probability outputs. Isotonic regression is applied to the final detection score to further calibrate the score. The decision threshold is set on the validation set based on a given false-positive-rate, which keeps the score within the SOC operational constraints.

4.5. Benchmark Architectures

To evaluate the effectiveness of the proposed TL-SOC framework, several intrusion detection architectures were implemented and used as benchmark models.

Baseline Autoencoder–Classifier Pipeline. An XGBoost classifier and a multilayer perceptron autoencoder (MLP-AE) are combined in the first baseline. Because autoencoder-based anomaly detection models learn compact representations of typical network traffic and detect anomalies through reconstruction errors, they are frequently used in intrusion detection systems [3, 10, 11]. In this baseline, the autoencoder produces an anomaly score, while XGBoost provides supervised classification capability.

CTAE with the XGBoost classifier. The second architecture replaces the MLP autoencoder with a CTAE, enabling the model to capture both local feature interactions and global dependencies within network traffic features. Because deep neural architectures can learn complex traffic representations, they have demonstrated strong performance for network anomaly detection [2, 4]. The anomaly score produced by the CTAE model is then combined with the XGBoost classifier.

Hybrid Multi-Model Fusion Architecture. The third architecture introduces a hybrid detection strategy combining two anomaly detection models: a CTAE and a GNN-AE. In [5, 6, 14], studies have shown that hybrid intrusion detection systems incorporating multiple learning models improve the robustness and accuracy of detection. In this configuration, the anomaly scores generated by these models are combined with the XGBoost probability through a weighted fusion mechanism.

4.6. Evaluation Protocol

In order to determine the reliability of the proposed TL-SOC framework under actual deployment circumstances, extra tests were performed in out-of-distribution (OOD) conditions. In these tests, detection model deployment is followed by the emergence of new attack patterns and these flows are then combined with 95% benign traffic in order to simulate operational SOC environments.

Zero-day attacks. Zero-day attack detection is based on a hold-out protocol in which certain attack types are kept completely out of the training phase and are brought in only at the testing phase. In this paper, attack types like *Infiltration* and *Botnet* are excluded from the training dataset and used only at the testing phase. In order to mimic realistic SOC traffic distributions.

APT-like attacks. To simulate APT-like scenarios, at the time of selection, the traffic flow is assumed to be an advanced persistent threat as a result of low traffic volumes and long time intervals. First, the TL-SOC model has not been retrained on these OOD samples to evaluate its ability to generalize to new unseen attack behaviors without any iteration. Although actual SOC systems may utilize some form of continual learning or updates, this procedure aims to determine the intrinsic generalization capability of the model without updates.

In terms of evaluation scenarios, the same time-based division of the data is used, ensuring that training, validation, and test phases do not overlap, and preventing any form of information leakage. This also guarantees the same set of known versus unknown experiments.

4.7. Data Splitting Strategy

To ensure a fair and applicable evaluation of the proposed TL-SOC framework, two data allocation strategies are considered: stratified random allocation and chronological allocation.

The distribution of classes across the training, validation, and test sets is preserved by the first tactic, stratified allocation. Although this method typically produces excellent results, it may induce an optimistic bias, especially in intrusion detection circumstances where the test and training data may contain identical attack patterns. A chronological time-series split is used to overcome this restriction. Early network traffic is used for training, and later traffic is left for testing and validation. The dataset is divided based on time order. This protocol ensures strict separation between training, validation, and test sets, thereby preventing any form of data leakage and better reflecting real-world Security Operations Center (SOC) conditions, where models must detect evolving and previously unseen threats.

The transition from a stratified to a time-series split significantly increases the difficulty of the task, as the model is required to generalize under distribution shift and temporal drift. Both evaluation protocols are reported in this work to highlight the trade-off between idealized performance and realistic deployment conditions.

4.8. Evaluation Metrics

The performance of TL-SOC was evaluated using standard classification metrics widely adopted in intrusion detection research. Let TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. Precision and Recall measure the ability of the system to correctly detect malicious traffic:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

The F1-score summarizes the trade-off between precision and recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{19}$$

Because SOC environments require strict control of false alerts, the false positive rate (FPR) is explicitly monitored:

$$FPR = \frac{FP}{FP + TN} \tag{20}$$

In TL-SOC, the decision threshold is calibrated on the validation set to satisfy a predefined FPR constraint, ensuring compatibility with operational SOC requirements. To assess the models' ranking ability independently of the decision threshold, ROC-AUC and PR-AUC values are also presented. ROC-AUC measures the overall separability between legitimate and malicious traffic, while PR-AUC is particularly informative in highly unbalanced contexts. And finally, robustness was evaluated under out-of-distribution (OOD) conditions simulating realistic SOC scenarios. Two OOD settings were considered: zero-day attacks, where specific attack categories are excluded from training and evaluated only during testing, and APT-like attacks, which simulate stealthy low-rate and long-duration communication patterns. In both cases, the evaluation follows a SOC-oriented distribution where attacks are mixed with 95% benign traffic, and the models are not retrained on these OOD samples.

5. Experimental Results

To provide a clear and rigorous evaluation of the proposed TL-SOC framework, the experimental analysis is organized into two complementary settings.

1. The first setting corresponds to an in-dataset evaluation, where the model is trained and tested on CICIDS-2017 under different protocols, including stratified split, time-series split, and out-of-distribution (OOD) scenarios.

2. The second setting corresponds to a cross-dataset evaluation, where the model is trained on CICIDS-2017 and tested on CSE-CIC-IDS2018 without retraining, in order to assess generalization under distribution shift.

5.1. Detection Performance on Known Attacks

We first evaluate TL-SOC under a controlled in-dataset setting using the CICIDS-2017 dataset, which represents the baseline evaluation scenario. A stratified split was be applied to separate the dataset's 2,574,264 network flows into training, validation, and test sets. The trained model's generalization performance is evaluated using the final test set, which comprises 514,853 flows.

Table 3 reports the detection performance of TL-SOC on the test set using standard intrusion detection metrics.

Table 3: Detection performance of the proposed TL-SOC framework on the CICIDS-2017 test set.

Prec.	Rec.	F1-score	FPR	ROC-AUC	PR-AUC
0.9963	0.7444	0.8521	5.52×10^{-4}	0.9978	0.9883

The framework combines anomaly scores produced by the CTAE, structural anomaly scores generated by the GNN-AE, and classification probabilities obtained from the XGBoost post-filter. These signals are fused through a meta-learning decision layer that produces a unified detection score. The final decision threshold was calibrated on the validation set to satisfy a predefined false-positive constraint ($FPR_{target} = 5{\times}10^{-4}$). Under this configuration, TL-SOC achieves a precision of 0.9963, a recall of 0.7444, and an F1-score of 0.8521, while maintaining a very low false-positive rate of 5.52×10^{-4}. The model also achieves a ROC-AUC of 0.9978 and a PR-AUC of 0.9883, indicating strong discrimination between benign and malicious traffic. Additional indicators thus confirm the robustness of the model, with balanced accuracy of 0.8719 and a Matthews correlation coefficient (MCC) of 0.8397.

The confusion matrix obtained on the test set is reported in Table 4. The model correctly classifies the vast majority of benign flows while maintaining strong detection capability for malicious traffic. Only a small number of benign flows are incorrectly flagged as attacks, confirming the effectiveness of the FPR-constrained calibration strategy.

Table 4: Confusion matrix of TL-SOC on the CICIDS-2017 test set.

	Pred. Benign	Pred. Attack
Benign	429440	237
Attack	21774	63402

These results suggest that TL-SOC is able to achieve strong detection performance while maintaining strict control over false positives. This balance is especially important in SOC environments, where too many false alarms can quickly make analysts tired of responding to them and make it harder for them to respond to real threats. By limiting unnecessary alerts, the proposed approach helps ensure that detected events remain actionable and relevant in practice.

5.2. Impact of Data Splitting Strategy

In this section, we will compare stratified and time-series evaluation protocols to further assess robustness under realistic conditions. Table 5 shows a comparison of TL-SOC's performance under stratified versus time-series evaluation protocols.

For the stratified split, the performance is considerably better (F1-score = 0.8521) because the training and test sets are assumed to be drawn from the same distribution. Nonetheless, it is not clear how well the model generalizes, as there could be a data leakage, meaning the same attack patterns are present in both the training and testing sets. On the other hand, the time-series split, where the model is assessed using temporally separated and previously unseen attack patterns, yields a lower, yet more realistic performance (F1-score $\approx$ 0.5179). This evaluation method is more suitable in the context of real-world SOC environments that are subject to change, where attack patterns shift, and the model must generalize to new patterns in a different distribution. This is the reason why we worked on analyzing the protocols established to guide realistic expectations in intrusion detection systems. Stratified evaluation is focused on attaining the best detection performance, while time-series evaluation is focused on robustness and addressing the needs of real-world applications.

Table 5: Comparison of evaluation protocols

Protocol	Precision	Recall	F1-score	FPR
Stratified Split	0.9963	0.7444	0.8521	5.5×10^{-4}
Time-Series Split	0.8303	0.3763	0.5179	0.077

Despite the increased difficulty of the time-series setting, the TL-SOC framework maintains competitive performance, demonstrating its practical relevance for deployment in real-world SOC environments.

5.3. FPR-Constrained Threshold Calibration

Since SOC environments require strict control of false alarms, we analyze the impact of FPR-constrained threshold calibration within the in-dataset setting; maintaining a low false-positive rate is essential to avoid overwhelming security analysts with excessive alerts. For this reason, TL-SOC adopts a threshold calibration strategy that explicitly constrains the false-positive rate (FPR). The decision threshold is selected on the validation set to satisfy a predefined target FPR while preserving detection capability.

Let $s(x)$ denote the final detection score produced by the TL-SOC meta-fusion layer. A decision threshold τ is chosen such that the empirical false-positive rate on the validation set satisfies

$$FPR(\tau) \leq FPR_{target}. \tag{21}$$

In our experiments, the target value was set to $FPR_{target} = 5{\times}10^{-4}$. Using this calibration procedure, the selected threshold was $\tau = 0.9789$, resulting in a validation false-positive rate of 2.28×10^{-4} while maintaining a recall proxy of 0.6819. To analyze the trade-off between detection capability and false-positive control, additional thresholds were evaluated for several FPR targets. As the allowed FPR increases, the decision threshold decreases and the recall improves accordingly. For example, relaxing the constraint from 10^{-4} to 10^{-3} increases the recall proxy from 0.4088 to 0.7478.

Table 6: Threshold calibration under different FPR constraints.

FPR Target	Threshold τ	Recall Proxy
10^{-4}	0.9971	0.4088
5×10^{-4}	0.9789	0.6819
10^{-3}	0.9467	0.7478
2×10^{-3}	0.9176	0.8700
5×10^{-3}	0.6082	0.9117

This calibration mechanism ensures that TL-SOC remains compatible with operational SOC requirements by explicitly controlling the alert rate while preserving strong detection capability. Figure 2 illustrates the distribution of the TL-SOC anomaly scores on the validation set together with the selected decision threshold.

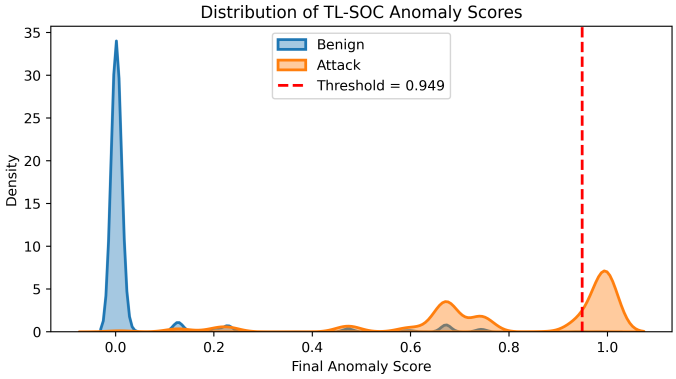

Figure 2: Distribution of TL-SOC anomaly scores for benign and attack traffic with the calibrated FPR-constrained decision threshold.

As shown in the figure, benign traffic scores remain tightly concentrated near zero, indicating that they are well reconstructed and consistent with the learned normal behavior, while malicious flows exhibit significantly higher scores, reflecting stronger deviations from normal patterns. The calibrated threshold is positioned between these two distributions, allowing a clear separation between benign and malicious traffic. This enables effective detection of anomalies while minimizing overlap between classes, while maintaining an extremely low false-positive rate essential for reliable and actionable alerts in SOC environments.

5.4. Ablation Study of TL-SOC Components

To assess the contribution of each module in the proposed TL-SOC framework, an ablation study was conducted by evaluating several configurations of the pipeline. The experiments compare the performance of the individual components (CTAE, GNN-AE, and XGBoost) with the final meta-fusion model under the same FPR-constrained calibration ($FPR_{target} = 5 \times 10^{-4}$).

Figure 3 illustrates the performance of the different configurations. Individually, the anomaly-based models remain weak. CTAE achieves a recall of 0.0054 and an F1-score of 0.0108, while GNN-AE improves slightly to an F1-score of 0.1800. In contrast, XGBoost yields strong performance (F1 = 0.8210, ROC-AUC = 0.9963). The best results are obtained with the full TL-SOC framework, which reaches F1 = 0.8521, precision = 0.9963, recall = 0.7444, ROC-AUC = 0.9978, and PR-AUC = 0.9883, while maintaining a very low false-positive rate.

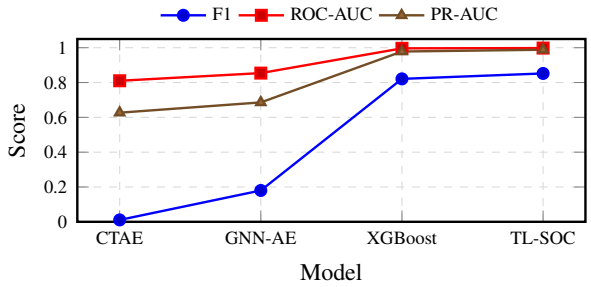

Figure 3: Performance comparison of TL-SOC components and the final meta-fusion model.

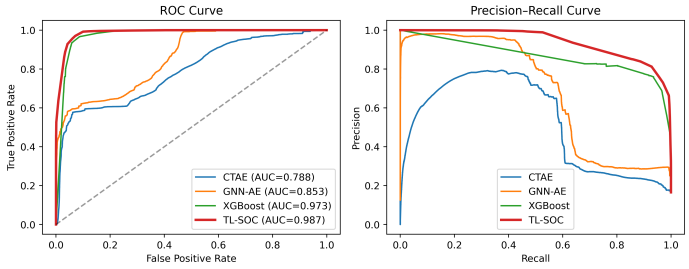

Figure 4: ROC and Precision–Recall curves comparing the individual components of the TL-SOC framework.

These results demonstrate that the different components of TL-SOC provide complementary detection signals. While anomaly-based models capture deviations from normal traffic behavior, the supervised classifier enhances discrimination between benign and malicious flows. Their integration through the meta-fusion layer leads to a more robust and accurate detection system.

5.5. Benchmarking of Detection Architectures

To assess the effectiveness of the proposed TL-SOC framework, a comparative evaluation was conducted against several alternative architectures with increasing modeling complexity. These architectures progressively integrate different detection mechanisms, enabling a systematic analysis of the contribution of each component in the final decision pipeline. The first architecture combines a reconstruction-based anomaly detector with a supervised classifier. A MLP-AE models normal traffic through reconstruction error, while an XGBoost classifier performs supervised classification of benign and malicious flows. The second architecture replaces the MLP-AE with a more expressive CTAE, enabling the model to capture more complex feature interactions in network traffic data. The third architecture introduces structural anomaly modeling by incorporating a GNN-AE. In this setup, anomaly scores produced by theses models are combined using a fixed weighted fusion strategy. Finally, the proposed TL-SOC framework replaces this fixed fusion mechanism with a meta-learning decision layer that adaptively combines the outputs of the detection modules, allowing the system to exploit the complementary strengths of anomaly detection and supervised learning.

Table 7: Benchmark comparison of the evaluated architectures on known attacks and out-of-distribution scenarios.

Arch	Known (F1)	APT (F1)	Z-Day (F1)	FPR
Arch1	0.901	0.543	0.059	6.21×10^{-4}
Arch2	**0.936**	0.226	0.088	9.82×10^{-3}
Arch3	0.797	0.159	0.199	5.68×10^{-4}
TL-SOC	0.852	0.247	**0.591**	5.52×10^{-4}

The benchmark comparison is reported using F1-score to ensure consistent evaluation across both in-distribution and out-of-distribution scenarios. Figure 5 summarizes the performance of the evaluated architectures under both in-distribution and out-of-distribution scenarios.

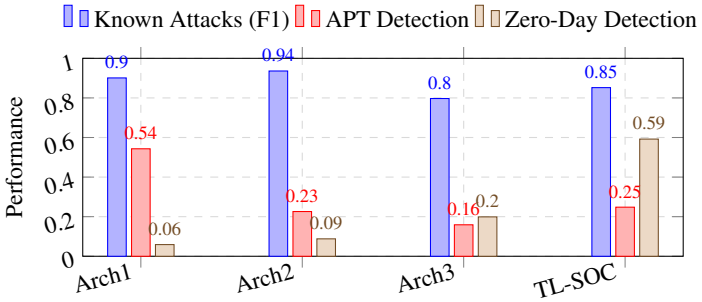

Figure 5: Benchmark comparison of the evaluated architectures.

These results highlight a clear trade-off between maximizing performance on known attacks and ensuring robustness to unseen threats. While Arch2 achieves the highest performance on known attacks (F1 = 0.936), it exhibits a higher false-positive rate (9.82×10^{-3}) and a significant performance drop in zero-day detection (F1 = 0.088). In contrast, TL-SOC maintains competitive performance on known attacks ($F1 = 0.852$) while substantially improving detection capability for zero-day attacks ($F1 = 0.591$) under strict false-positive constraints. This behavior is particularly important in SOC environments, where minimizing false alerts and maintaining reliable detection under evolving attack patterns are critical. By combining anomaly representation learning, structural modeling, and adaptive meta-learning fusion, TL-SOC effectively captures complementary detection signals, leading to improved resilience against previously unseen cyber threats.

Table 8: Comparison with state-of-the-art intrusion detection approaches on CICIDS-2017.

Method	Year	F1-score	ROC-AUC
Deep Autoencoder [2]	2018	0.85	0.97
CNN-based IDS [3]	2020	0.89	0.98
Hybrid DL Model [5]	2021	0.91	0.99
Ensemble IDS [6]	2024	0.93	0.995
Decision Fusion IDS [7]	2024	0.92	0.994
TL-SOC (proposed)	2025	0.852	**0.9978**

These results demonstrate that, although some state-of-the-art approaches achieve higher performance on known attacks, they generally remain accuracy-driven and do not explicitly address the operational constraints of SOC environments. In particular, they lack mechanisms to control false-positive rates and to ensure robustness under distribution shifts. In contrast, TL-SOC adopts a decision-centric architecture that integrates anomaly representation learning and supervised classification within a unified framework. The proposed meta-learning-based fusion layer plays a key role by adaptively combining heterogeneous detection signals, enabling the model to exploit their complementary strengths while maintaining calibrated and reliable decision-making. This design allows TL-SOC to achieve a favorable trade-off between detection performance and operational reliability, characterized by low false-positive rates, improved generalization to unseen threats, and enhanced interpretability. As a result, TL-SOC is particularly well-suited for real-world SOC deployment, where robustness and alert precision are critical.

5.6. *Robustness to Out-of-Distribution Attacks*

To further evaluate the robustness of TL-SOC beyond standard in-distribution conditions, we consider out-of-distribution (OOD) scenarios that simulate realistic SOC environments where new attack patterns emerge after deployment. Two types of OOD threats were considered: *APT-like attacks* and *zero-day attacks*. The evaluation datasets were generated by mixing OOD attack flows with benign traffic to reproduce SOC traffic distributions. Unless otherwise specified, the mixture contains 95% benign traffic and 5% OOD attack flows.

APT-like attacks. The APT scenario includes previously unseen attack patterns generated from multiple sources, including synthetic perturbations and unknown real attack samples. Under the SOC-like distribution (95% benign traffic), TL-SOC maintains strong discrimination capability with ROC-AUC values reaching up to 0.9834 and PR-AUC up to 0.8148. Despite the strict FPR constraint used during threshold calibration, the model is still able to identify anomalous traffic patterns associated with advanced threats.

Zero-day attacks. Zero-day detection was evaluated using attack categories that were completely excluded from the training process. These attack flows were introduced only during testing and mixed with benign traffic to simulate realistic operational conditions. Under the same SOC-like distribution, TL-SOC achieves ROC-AUC values of approximately 0.92 and PR-AUC values up to 0.6. The model reaches recall values between 0.25 and 0.43 depending on the zero-day scenario, while maintaining a very low false-positive rate.

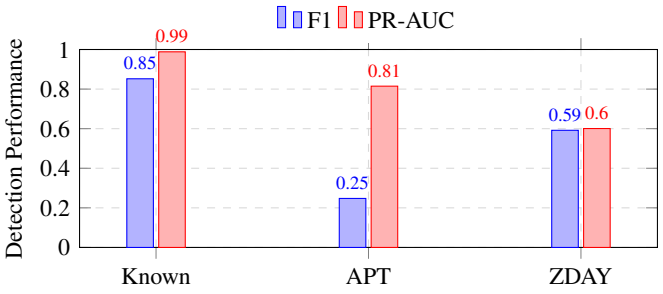

Figure 6: Comparison of TL-SOC detection performance across evaluation regimes.

Overall, these results demonstrate that TL-SOC maintains robust detection performance under distribution shifts, preserving low false-positive rates while effectively identifying previously unseen threats. The combination of anomaly-based modeling and supervised classification in a meta-learning fusion framework makes this behavior possible. This framework makes sure that performance stays stable even when the data distribution changes. In SOC environments, where reliable detection is necessary for operational deployment, this kind of robustness is very important.

5.7. Model Interpretability and Feature Analysis

To improve transparency and better understand the behavior of the proposed TL-SOC framework, an interpretability analysis was conducted using SHAP, which quantifies the contribution of each feature to the final decision score of the model.

We first analyze the importance of the meta-fusion inputs used by the final decision layer. The meta-model takes three signals and combines them: the CTAE's anomaly score (s_{ctae}), the GNN-AE's structural anomaly score (s_{gnn}), and the XGBoost post-filter's classification probability (p_{xgb}). The SHAP analysis shows that the XGBoost probability has the biggest impact on the final decision, followed by the graph-based anomaly score. The CTAE score is a secondary signal.

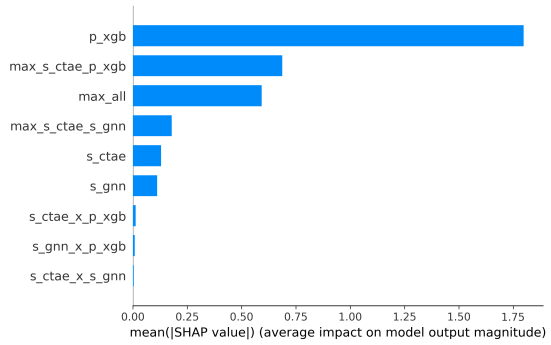

Figure 7: SHAP-based importance of the TL-SOC meta-fusion inputs.

We further analyze the importance of network traffic features in the XGBoost classifier to better understand the decision-making process of the proposed framework. The most influential variables include *Destination Port*, *Init_Win_bytes_backward*, *Subflow Bwd Bytes*, *Subflow Fwd Bytes*, and *Average Packet Size*. These characteristics appear to represent significant statistical and structural aspects of network flows, such as communication endpoints, the volume of two-way traffic, and packet size. Specifically, the *Destination Port* shows the service that is being targeted and can show unusual access patterns. The *Init_Win_bytes_backward* shows how TCP flow control works and can show unusual communication patterns. The same goes for "Subflow Bwd Bytes" and "Subflow Fwd Bytes," which show how much data is sent and received in both directions. This can help you understand traffic patterns that are not normal or are very heavy, which are often linked to bad behavior. The distribution of packet payloads, which can vary greatly between benign and attack traffic, is further described by the *Average Packet Size*. The prominence of these features indicates that the model relies on meaningful and interpretable traffic characteristics rather than spurious correlations. This is particularly important in cybersecurity applications, where reliable detection must be grounded in realistic network behavior. The SHAP beeswarm analysis further reveals consistent and coherent relationships between feature values and their contribution to the model output. For instance, high values of traffic volume-related features, such as forward and backward byte counts, tend to increase the probability of malicious classification, reflecting abnormal communication intensity commonly observed in attacks such as denial-of-service or data exfiltration. Conversely, lower or more regular values are generally associated with benign traffic patterns.

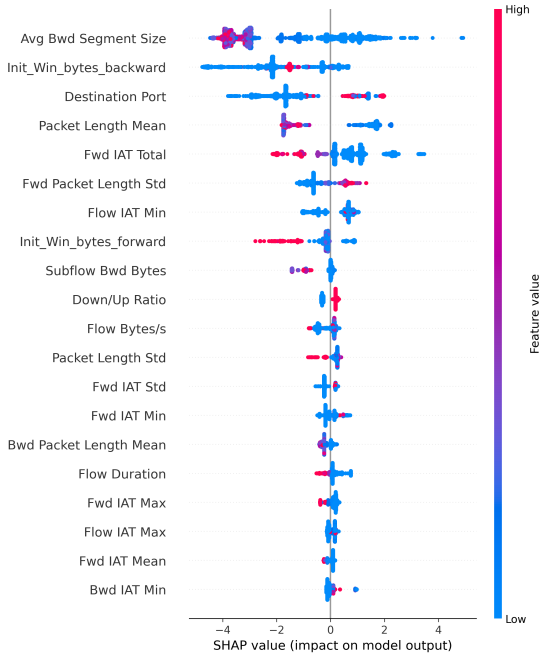

Figure 8: SHAP beeswarm plot illustrating the distribution and directional impact of feature values on the XGBoost classifier.

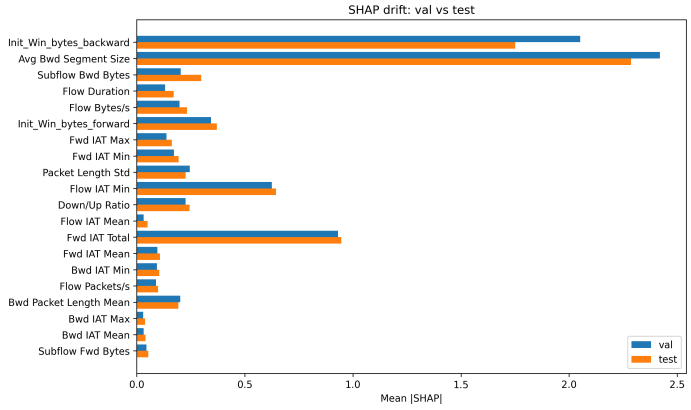

Figure 9: Comparison of mean SHAP values between validation and test sets. The consistency of feature importance across datasets indicates that the model captures generalizable patterns rather than dataset-specific artifacts.

This consistency highlights the stability of the learned decision patterns and supports the interpretability and reliability of the model. It also indicates that TL-SOC captures domain-relevant

features aligned with typical intrusion behaviors, facilitating analyst understanding. The SHAP drift analysis further shows that feature importance remains consistent across validation and test sets, indicating that the model learns generalizable patterns rather than dataset-specific artifacts. This stability is essential for reliable detection under distribution shifts in SOC environments.

5.8. Cross-Dataset Generalization

In addition to the in-dataset evaluation, we conduct a cross-dataset experiment to assess the generalization capability of TL-SOC under strong distribution shift. To further evaluate the generalization capability of the proposed TL-SOC framework, cross-dataset experiments were conducted by training the model on the CICIDS-2017 dataset and testing it on the CSE-CIC-IDS2018 dataset without any retraining. This experimental setup simulates realistic deployment conditions in Security Operations Centers (SOCs), where intrusion detection systems must operate under distribution shifts and encounter previously unseen traffic patterns. Despite the inherent differences between datasets in terms of traffic characteristics, feature distributions, and attack types, the TL-SOC framework maintains stable and reliable performance. The results demonstrate that the proposed decision-centric hybrid architecture effectively generalizes across heterogeneous environments while preserving strict control over false-positive rates. As anticipated, domain shift results in a decline in recall. Nonetheless, the model maintains a low false-positive rate and excellent precision, guaranteeing the production of trustworthy and useful alerts. In SOC situations, where reducing false alarms is essential to prevent overburdening security analysts, this behavior is especially crucial.

Table 9: Cross-dataset evaluation of TL-SOC (Train: CICIDS-2017, Test: CSE-CIC-IDS2018)

Metric	**TL-SOC**
Precision	0.96
Recall	0.42
F1-score	0.58
FPR	7.2×10^{-4}
ROC-AUC	0.93
PR-AUC	0.64

6. Discussion

We've thoroughly tested our TL-SOC system in both perfect and more typical real-world situations and have learned a lot about how it works and how useful it could be for Security Operations Centers.

One important thing we found is that combining different methods for finding intrusions works best. When we took apart the system and tested just the CNN-Transformer Autoencoder (CTAE) and Graph Neural Network Autoencoder (GNN-AE) anomaly detectors, we found they aren't very good at finding attacks if you also need to be sure they won't give many false alarms.This restriction is usual in unsupervised anomaly detection; even though the method normal modeling is exemplary in its ability, the evidence for decision reliability is largely insufficient. Extreme Gradient Boosting (XGBoost), a supervised classifier, is excellent at distinguishing between normal and bad traffic, though is susceptible to changes in data distribution. The proposed meta-learning fusion layer balances all of these attributes by combining the ability to adapt to new patterns and the ability to make supervised classifications. It also indicates that the primary utility of TL-SOC is not in the performance of a single model, but in a high-functioning interplay of many models. One of the most important findings is regarding the false-positive-rate-constrained calibration. In contrast to most intrusion detection systems, which optimize performance metrics that are either accuracy or F1, TL-SOC controls trade-offs along a decisive axis. The results indicate that when the false positive rate is kept very low, the model is adjusted in a significant and reliable way, but at the cost of recall. In this constrained environment, TL-SOC is able to deliver a high level of certainty in its alerts through a commendable balance of detection ability and precision. This is especially pertinent to SOC environments, where analysts can be overwhelmed by floods of false alerts, negatively impacting operational efficiency (our validation showed that valid entries there strongly predicted false alerts). This study provides insight into the range of potential metrics for measuring the performance of an intrusion detection system. Specifically, the time-series and out-of-distribution (OOD) experimental settings highlight performance and generalization trade-offs. As expected, the more challenging the evaluation (i.e., temporal and OOD settings), the more performance declined. The trade-offs were consistent across all models, demonstrating the value of the settings in capturing the complexity of real-world scenarios. Of particular note, TL-SOC is less impacted by the zero-day attacks due to the signalling and detection process being more robust. While hybrid designs support an improvement, the detection of (APT) Advanced Persistent Threat-like low signal attacks still remains challenging, as identified by the low recall. These low signal attacks highlight the difficulty of identifying subtle irregularities in otherwise normal communication. The interpretability analysis provides more insight into the framework, with SHAP results confirming XGBoost to be the dominant signal. This suggests that the outcomes are more dependent on the robust anomaly-based components rather than the standalone signal, and shows that the role of anomaly detection in TL-SOC is primarily to support integrity and resilience to distribution changes. In addition, the uniformity of feature importance for the validation and test sets implies the model stable and generalizable captures generalizable and stable traffic patterns and artifacts. Still, SHAP explanations are also interpretable post hoc and lack causally, which limits impact of explanation for security sensitive use cases. Evaluation itself presents another valuable result. Coupled with the important confirmation that random partitioning strategies can inflate performance estimates due to unrecognized temporal leakage, we observe significantly large differences in the case of stratified versus not stratified splits. On the contrary, chronological evaluation depicts a real world scenario where a model operated under an evolving traffic distribution with shifting traffic patterns across time. This phenomenon indicates the importance of consistent temporal evaluation in intrusion detection. Finally, the cross-dataset evaluation demonstrates that TL-SOC is able to sustain requirements for a better stable precision and satisfactory false positive rate with lower recall on

unseen datasets, which is the most desirable detection operational preference. This abnormality prioritizes the generation of reliable, actionable alerts less aggressively. Even though the results look good, we need to be honest about what this work can't do. We tested our system with data everyone can get to, and that data might not show all the different and complicated things that happen on actual networks. Also, because of its combined approach, this more complex system requires more computing power than simpler ways to find attacks. While this more complicated design makes it perform steadily and dependably, it could be a problem when dealing with a huge amount of network activity. So, to really understand complicated, multiple-step attacks, future work will focus on making the system faster, testing it with massive amounts of data from working networks, and using methods that analyze how things change over time. To summarize, the research findings are positive. Integrated operational boundaries, hybrid modeling, and calibrated decision fusion enable an intrusion detection decision-centric framework to improve the reliability, robustness, and practicality of these systems within modern SOC environments.

7. Conclusion and Future Perspectives

We present TL-SOC, a way of finding intrusions that's built for how Security Operations Centers (SOCs) work, and it's all about making decisions. It combines learning how normal things look (anomaly representation learning), modelling how different pieces of the system relate to each other (graph-based structural modeling), and standard 'labelled' categorization (supervised classification) into a single process for deciding if something is an attack. This is done by smartly combining all of these using a method called meta-learning, and importantly, it's done with a really tight restriction on how many false alarms you'd get. The experiments we ran show TL-SOC is good for actual use in a SOC because it gets a lot of attacks and doesn't have many false positives. It's also more reliable in tricky situations: looking at data over time, when faced with attacks that are unlike anything seen before (like Advanced Persistent Threats and zero-day exploits), and when used on datasets that are different from the ones it was trained on. Plus, we've used a sophisticated method based on SHAP values to explain both which aspects of the data are most important, and how the system is making its decisions. This makes it easier to understand and for analysts to have confidence in it. These results show how important it is to build intrusion detection that is not only accurate, but also strong, able to work in lots of different situations, and will work as expected in a real SOC. TL-SOC uses understandable AI, labelled learning, flexible combination of methods, and a way to identify unusual activity to give a balanced and you can count on it system for modern SOCs. For the future, we're going to test it on a lot more, and much more varied, real-world data. We'll also work on making it run faster for very high volumes of information, and add ways to understand attacks that happen in stages and change over time.

References

[1] T. Bass, "Intrusion detection systems and multisensor data fusion," Communications of the ACM, **43**, 99–105, 2000, doi:10.1145/332051.332079.

[2] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," IEEE Access, **6**, 48231–48246, 2018, doi:10.1109/ACCESS.2018.2863036.

[3] S. Zavrak, M. İskefiyeli, "Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder," IEEE Access, **8**, 108346–108358, 2020, doi:10.1109/ACCESS.2020.3001350.

[4] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, K. J. Kim, "A survey of deep learning-based network anomaly detection," Cluster Computing, **22**, 949–961, 2019, doi:10.1007/s10586-017-1117-8.

[5] C. Liu, Z. Gu, J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," IEEE Access, **9**, 75729–75740, 2021, doi:10.1109/ACCESS.2021.3082147.

[6] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, A. U. Rehman, "Enhancing intrusion detection: a hybrid machine and deep learning approach," Journal of Cloud Computing, **13**, 123, 2024, doi:10.1186/s13677-024-00685-x.

[7] R. Ahmad, I. Alsmadi, "Data fusion and network intrusion detection systems," Cluster Computing, **27**, 7493–7519, 2024, doi:10.1007/s10586-024-04365-y.

[8] Y. Xue, J. Pan, Y. Geng, Z. Yang, M. Liu, R. Deng, "Real-Time Intrusion Detection Based on Decision Fusion in Industrial Control Systems," IEEE Transactions on Industrial Cyber-Physical Systems, **2**, 143–153, 2024, doi:10.1109/TICPS.2024.3406505.

[9] I. Lotfi, M. Mandar, "Review of Detection and Prevention Techniques for Cyberattacks in SOCs: State of the Art and Future Challenges," in 2025 International Conference on Circuit, Systems and Communication (ICCSC), 1–6, 2025, doi:10.1109/ICCSC66714.2025.11135218.

[10] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, "Autoencoder-based network anomaly detection," in 2018 Wireless Telecommunications Symposium (WTS), 1–5, 2018, doi:10.1109/WTS.2018.8363930.

[11] M. Said Elsayed, N.-A. Le-Khac, S. Dev, A. D. Jurcut, "Network Anomaly Detection Using LSTM Based Autoencoder," in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet '20, 37–45, Association for Computing Machinery, New York, NY, USA, 2020, doi:10.1145/3416013.3426457.

[12] H. Rajadurai, U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," Neural Computing and Applications, **34**, 15387–15395, 2022, doi:10.1007/s00521-020-04986-5.

[13] R. Lazzarini, H. Tianfield, V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," Knowledge-Based Systems, **279**, 110941, 2023, doi:https://doi.org/10.1016/j.knosys.2023.110941.

[14] E. U. H. Qazi, M. H. Faheem, T. Zia, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," Applied Sciences, **13**(8), 2023, doi:10.3390/app13084921.

[15] A. Ayantayo, A. Kaur, A. Kour, X. Schmoor, F. Shah, I. Vickers, P. Kearney, M. M. Abdelsamea, "Network intrusion detection using feature fusion with deep learning," Journal of Big Data, **10**, 167, 2023, doi:10.1186/s40537-023-00834-0.

[16] C. Xu, J. Shen, X. Du, "A Method of Few-Shot Network Intrusion Detection Based on Meta-Learning Framework," IEEE Transactions on Information Forensics and Security, **15**, 3540–3552, 2020, doi:10.1109/TIFS.2020.2991876.

[17] T. Zoppi, M. Gharib, M. Atif, A. Bondavalli, "Meta-Learning to Improve Unsupervised Intrusion Detection in Cyber-Physical Systems," ACM Trans. Cyber-Phys. Syst., **5**(4), 2021, doi:10.1145/3467470.