From ML to LLM: A Comparative Analysis of Context-Aware Zero-Day Attack Detection

Moumita Shib*, Shaily Kabir†, Mosarrat Jahan*, and Upama Kabir*
*Department of Computer Science & Engineering, University of Dhaka, Dhaka, Bangladesh
†Lab for Uncertainty in Data and Decision Making (LUCID),
School of Computer Science, University of Nottingham, Nottingham, UK
Email:moumita-2022318426@cs.du.ac.bd, shaily.kabir5@nottingham.ac.uk, {mosarratjahan,upama}@cse.du.ac.bd

***Abstract*—Detection of zero-day attacks, those exploiting previously unknown vulnerabilities, is essential for mitigating severe cybersecurity risks and preventing large-scale damage. Conventional intrusion detection systems (IDSs) remain inadequate for such attack because they rely on known signatures or behavioural models that often lack stability. Although machine learning (ML) and deep learning (DL) approaches improve generalization, ML methods frequently fail to capture unseen behaviours, while DL models are sensitive to distribution shifts and limited in modelling long-range context. Neural networks can learn discriminative traffic representations, yet under severe distribution shifts or rare attack patterns they may still become overconfident and miss low-frequency zero-day events. To address these limitations, hybrid models have been proposed that fuse traditional detection signals with contextual scoring learned from neural networks to enhance robustness. This paper mainly focuses on hybrid designs incorporating IDSs with deep neural networks (DNNs) or large language models (LLMs) and evaluates their zero-day attack detection performance against state-of-the-art baselines across three real-world datasets—UNSW-NB15, TON-IoT, and CIC-Collection—differing in label structure, class imbalance, attack diversity, and zero-day evaluation settings. Results show that all hybrid models maintain comparatively stable performance on UNSW-NB15 and TON-IoT, whereas their detection capability deteriorates markedly on CIC-Collection, reflecting a significant collapse under its more challenging distributional characteristics.**

***Index Terms*—Network security, intrusion detection system (IDS), large language model (LLM), deep neural network (DNN)**

I. Introduction

The rapid advancement of information technology, coupled with the widespread use of the Internet, has led to a substantial increase in network traffic across enterprise, cloud, and other digital infrastructures. At the same time, cyberattacks have grown more sophisticated, particularly with the extensive deployment of Internet of Things (IoT) devices [1]. This expansion has significantly broadened the attack surface and increased the difficulty in securing modern systems [2]. Although security mechanisms such as Firewalls offer partial protection, they remain insufficient against zero-day attacks that exploit previously unknown vulnerabilities [3], [4]. Traditional intrusion detection systems (IDSs), typically categorized into anomaly-based (ABD) and signature-based (SBD) approaches [5], provide defensive support against a range of known attacks; however, both approaches exhibit inherent limitations in detecting zero-day attacks, manifesting as novel and previously unseen vulnerabilities [4], [6].

Machine learning (ML), deep learning (DL), and artificial neural networks (ANNs) have been widely explored to enhance intrusion detection by learning patterns from network traffic and log data [7]. However, many of these models struggle to generalize beyond their training distributions and capture more contextual information [2]. Moreover, under severe distribution shifts or rare attack patterns, neural network based detectors often become overconfident and unable to separate low-frequency zero-day events, despite their ability to learn discriminative traffic representations [8]. To mitigate these limitations, hybrid models have been proposed that combine the robustness of traditional IDSs with neural network-derived contextual scoring, enabling more holistic, context-aware intrusion analysis [2]. Additionally, large language models (LLMs), originally developed for natural language processing [9], have demonstrated strong capabilities in complex reasoning, contextual understanding, and cross-domain generalization, although they are often susceptible to hallucinations [10]. These advances have opened new opportunities for cybersecurity applications, particularly in zero-day threat detection [9], [11].

In this paper, we develop a hybrid intrusion detection pipeline that combines traditional IDSs with neural network-based contextual scoring to improve the zero-day robustness across diverse data settings. A core component of the proposed framework is a zero-day-oriented feature selection method based on Equilibrium Optimization (EO) [12], which extracts a compact and optimal subset of features from a dataset. We integrate a feedforward deep neural network (DNN) [13], a GPT-2 flow detector [14], and a Qwen3-based detector [15] over the EO-selected features, and construct hybrid attack detectors by fusing neural network- and TDS-derived scores to yield a final detection score.

The resulting hybrid models are further systematically evaluated with respect to three heterogeneous network datasets: UNSW-NB15 [16], TON-IoT [17], and CIC-Collection [18], differing in label granularity, class imbalance, attack diversity, and distribution shift and their performance are assessed using F1-score [1] and Zero-Day-Recall (zero-day detection rate) [19]. Experimental results show that the hybrid models maintain stable performance on UNSW-NB15 and TON-IoT,

having better represented attack families and zero-day samples, but degrade sharply on CIC-Collection due to its extreme class imbalance and scarcity of attack flows. We provide the entire code repository on GitHub[1].

The remainder of the paper is organized as follows. Section II reviews state-of-the-art IDSs. Section III outlines the proposed zero-day detection methodology along with details of hybrid models. Section IV discusses the required experimental setup and evaluation metrics, and Section V presents a comparative analysis of models' zero-day detection performance. Lastly, Section VI concludes the paper with future directions.

II. Literature Review

Traditional IDSs are commonly classified into SBD [20] and ABD [21] approaches. SBD IDSs typically achieve high accuracy in detecting known threats, but are ineffective against zero-day attacks in the absence of prior behavioural knowledge [3]. It is also true that the generation and maintenance of signatures for new rapidly evolving threats is expensive [22]. In contrast, ABD IDSs identify attacks by detecting deviations from normal behaviour, leading to a high rate of false positives [21]. To mitigate these limitations, hybrid IDSs have been proposed integrating SBD and ABD IDSs [2], offering a more robust solution for addressing known and unknown attacks [3], [23]. Although both IDSs increasingly incorporate ML and DL methods to improve detection performance [20], [21], [7], identifying zero-day attacks remains challenging due to their limited ability to capture complex contextual relationships. Consequently, researchers are turning towards neural network models capable of capturing richer contextual information.

Existing literature mostly utilizes Bidirectional Encoder Representations from Transformers (BERT) to enhance the zero-day attack detection capabilities of IDSs. Yang and Peng [24] introduced a BERT-based IDS that converts network traffic into language like sequences, allowing feature extraction through NLP techniques. With the extracted features, the model performs classification using a bidirectional Gated Recurrent Unit (GRU) network to distinguish between normal and abnormal traffic. Evaluated on publicly available multi-source network-traffic datasets [21], the proposed system achieves approximately 95% Accuracy and 94% F1-score, outperforming conventional ML and DL models.

Swileh and Zhang [13] proposed a novel approach with NLP and a pre-trained BERT-base-uncased LLM to detect unseen or zero-day attacks in Software Defined Networks (SDN). This approach transforms network flow data into a language-interpretable format, uses random forest [7] for optimized feature selection, and analyzes multiple network flows for robust attack decisions. The proposed scheme is evaluated on InSDN dataset [25] comprising 7 attack types and achieves nearly 100% Accuracy for known and unknown attack settings.

Mirza et al. [6] proposed a ZDBERTa model combining zero-shot learning with a BERT-based framework to tackle zero-day attacks in vehicular networks. Here, network traffic is first converted to language-like inputs, processed by a transformer encoder to capture complex patterns, and then classified using semantic features and zero-shot learning to classify unseen attacks. Evaluated on the CICIoV2024 dataset [26] under two synthetic data-augmentation settings, ZDBERTa achieves nearly 100% Accuracy when Generative Adversarial Network (GAN) [6]-generated synthetic samples are used.

Binhulayyil et al. [1] developed CyBERT—an LLM-based classifier trained directly on raw network traffic bytes, avoiding manual feature extraction to speed up processing. Here, network traffic is encoded as byte sequences, embedded, processed by transformer layers, and finally classified. Using the Ripple20 dataset collected from a Smart Home lab (containing *normal* traffic and 19 *attack* types [1]) in a binary classification setup, CyBERT achieved 100% Accuracy, Precision, Recall, and F1-score, along with AUC-ROC and AUC-PRC of 1.0.

Alsuwaiket [19] introduced a ZeroDay-LLM model aiming to make LLM-based zero-day detection practical for real-time deployment in heterogeneous networks. The model uses a hybrid edge–cloud design, where lightweight and compressed BERT-like encoders analyze traffic at the edge, and a central transformer reasoning engine performs deeper contextual reasoning and classification. Experimented with 4 datasets including CICIDS2017 [21], NSL-KDD [27], UNSW-NB15 [16], and an IoT-custom dataset, ZeroDay-LLM shows great performance on multiple evaluation aspects, with a detection Accuracy of 97.8%, a zero-day attack detection rate of 95.7%, and real-time processing with an average latency of 12.3 ms.

From the perspective of LLM-based methods for detecting unseen threats, IDS Agent [11] focuses on IoT intrusion detection, addressing the limits of SBD/ABD IDSs on zero-day behaviour and the poor interpretability of ML-based systems. The approach formulates intrusion detection as an LLM-agent task, where a core LLM executes a reason–act loop to coordinate traffic preprocessing, multi-model classification, and final decision aggregation, supported by a memory base for contextual reasoning and explanation. The system employs several classical classifiers (e.g., RF, SVM, MLP, DT, KNN); when their outputs diverge, the agent retrieves additional attack knowledge via search/RAG and memory, then combines top-k classifier confidences with the retrieved context to produce a structured JSON decision and explanation. Evaluating the IDS agent on the ACI-IoT'23 [28] and CIC-IoT'23 [29] datasets, 97% of the F1-score was achieved for the first dataset while 75% for the second with 61% Zero-Day-Recall. However, its benchmarking scope is limited, and the multi-stage LLM pipeline may be computationally heavy and prone to over-alerting without stronger calibration.

III. Material and Methods

Figure 1 shows the proposed methodology for the zero-day attack detection.

[1]https://github.com/moumita-papercode/context-aware-zero-day-attack-detection/

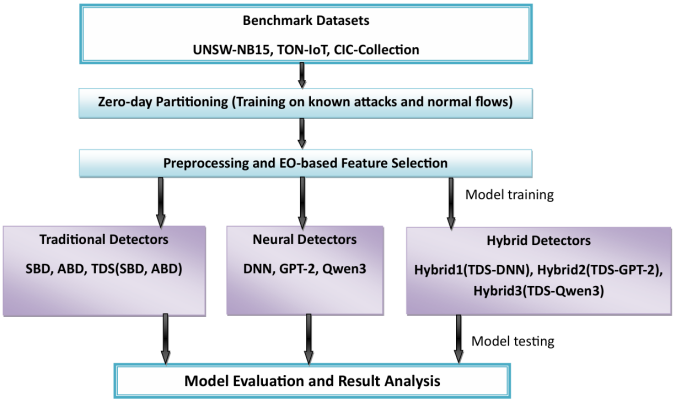


Fig. 1: Proposed methodology for zero-day attack detection.

A. Datasets for Model Evaluation

1) UNSW-NB15 [16]: It comprises UNSW-NB15_TRAIN for model training and UNSW-NB15_TEST for model evaluation. Records are grouped into two categories: *normal* and *attack*. In total, it contains 257,673 flows, including 4,362 flows belong to *attack* families (zero-day *attack* samples). From an initial pool of 39 candidate features, preprocessing yields a refined set of 31 numeric features.

2) TON-IoT [17]: It consists of network traffic from simulated IoT and IIoT environments. It has 211,043 flows in total, including 50,000 labeled as *normal* and 161,043 as *attack*. We randomly selected 20% of the attack flows as unseen 'zero-day' traffic. The final splits comprise 125,184 training flows, 26,825 validation flows, and 37,500 test flows, of which 20,571 correspond to zero-day attacks. After preprocessing, 37 features are retained from the original set of 39 features.

3) CIC-Collection [18]: It is substantially larger and more diverse dataset than UNSW-NB15 and TON-IoT, as it aggregates multiple sources of network intrusion detection traffic. It contains 9,167,581 flows in total, with 7,186,189 labeled as *normal* and 1,981,392 as *attack*. The *Web Attack* category (e.g., SQL injection, XSS, and HTTP-layer attacks) is designated as the zero-day attack family. For computational tractability, it is subsampled into 210,000 training flows, 45,000 validation flows, and 45,000 test flows. After preprocessing, the original 58 features are reduced to 33 numeric features.

B. Preprocessing and Feature Engineering

For each dataset, we applied preprocessing steps to enhance model performance and accuracy. From the original labels, a binary target variable was constructed, assigning '1' to *attack* and '0' to *normal* traffic. Further, redundant columns were removed to avoid trivial memorization, infinite values were corrected, and all remaining attributes were converted to numeric form. We applied 'LabelEncoder' [21] on the training set and mapped the validation and test values through the same encoder to prevent information leakage.

Constant-valued columns (those with at most one unique value in the training split) were removed, and highly correlated features (absolute correlation above 0.95) were filtered to reduce redundancy. All remaining features were scaled to [0,1] using a MinMax scaler [2]. The resulting normalized feature vectors were then provided to all detectors, ensuring a consistent input representation and preventing any test-time or zero-day information from leaking in the preprocessing stage.

C. Equilibrium Optimizer Based Feature Selection

We used Equilibrium Optimizer (EO) [12] to derive a compact yet discriminative feature subset from the preprocessed training split of each dataset. In this setting, each EO individual was a binary mask $\mathbf{m} \in \{0,1\}^d$ over d preprocessed features, with a population of 30 for 50 iterations. The fitness of each mask was calculated using an XGBoost classifier [30] trained solely on the training split with 3-fold cross-validation and class weighting. We computed mean recall, $\mathrm{Rec}(\mathbf{m})$ and mean F1-score, $\mathrm{F1}(\mathbf{m})$, combining them into a utility function $s(\mathbf{m}) = 0.4\,\mathrm{Rec}(\mathbf{m}) + 0.6\,\mathrm{F1}(\mathbf{m})$. The final fitness was defined as $\mathrm{fit}(\mathbf{m}) = \alpha\, s(\mathbf{m}) - \beta\, \frac{\|\mathbf{m}\|_0}{d} + \gamma$, with $\alpha = 0.8$ emphasizing detection performance, $\beta = 0.15$ penalizing larger subsets, and $\gamma = 0.05$ providing a small bias term; masks selecting no features were assigned a fitness of 0.

We obtained stable cross-validated fitness values of approximately 0.7683 for UNSW-NB15, 0.7942 for TON-IoT, and 0.8079 for CIC-Collection, resulting in final EO-selected subsets of 12, 13, and 8 features, respectively. These selected features served as the unified input space for all downstream detectors, providing an effective balance between model-complexity and zero-day detection performance.

D. Detection Models

1) Traditional Detection Scheme (TDS): The TDS comprised an SBD and an ABD IDSs operating directly on the EO-selected feature subset, with the rule-based SBD filtering intrusions prior to the application of the more flexible anomaly-based detector [31], [32]. The SBD module adapted the signature analysis phase algorithm [31], identifying requests violating expected protocol or credential patterns. In the flow-level setting, benign 'signatures' were learned by analyzing value ranges on low-level features. All flows labeled as *normal* in the training split were used to compute lower and upper bounds for each feature. During inference, a flow was marked as *suspicious* if the number of features falling outside the benign ranges exceeds a configurable threshold. The SBD then produced a binary score per flow indicating whether any signature violations occurred.

For anomaly detection, we used a contextual one-class nearest neighbors [32], enabling the ABD module to capture distance-based deviations from benign behaviour. The EO-selected features were standardized with a StandardScaler [7] fitted on the benign training flows, after which truncated singular value decomposition (SVD) [32] is applied to obtain a lower-dimensional representation. In this space, a k-nearest neighbors model (with $k = 2$) was used to classify *normal* and *attack* flows. The final TDS model was created using the maximum fusion [2] to the SBD and ABD outputs.

2) Deep Neural Network Detector (DNN): We trained a DNN model [13], where the input layer is zero-padded to 512 dimensions and successively standardized using a StandardScaler [7] fitted on the training split. The network comprised three hidden layers with 128 ReLU-activated units each, with dropout (0.5) applied after the second and third layers to reduce overfitting. The final layer was a single sigmoid neuron that produced a probability of the flow being malicious. The DNN was optimized using binary cross-entropy loss and the AdamW optimizer [13] with a fixed learning rate of 10^{-3}. During inference, the model produced a calibrated probability in the range [0,1] for each flow.

3) GPT-2 Flow Detector: We adapted the GPT-2 flow transformer based detector [14] to effectively capture higher-order feature interactions that may be under-utilized by a purely feedforward DNN. Each network flow was represented as a short sequence of EO-selected normalized features, generated using a sliding-window scheme with window length, $L = 32$ and stride, $s = 4$. We employed a compact GPT-2 configuration with a hidden layer size of 512, four transformer blocks, eight attention heads, and a dropout rate of 0.1 applied to both embeddings and attention. The model processed each sequence and produced hidden states for all positions. We aggregated the information by combining the last token and fed the resulting representation into a small feedforward classifier that produced a sigmoid attack probability. Training was performed using binary cross-entropy and the AdamW optimizer [13] with a learning rate of 10^{-4} and weight decay of 0.01.

4) Qwen3 Detector: We fine-tuned a compact instruction following LLM—Qwen3-0.6B [15]—as a lightweight IDS detector using LoRA (Low-Rank Adaptation)-based parameter-efficient fine-tuning approach [33]. We used AdamW optimization with a learning rate of 2×10^{-4}, cosine scheduling, a per-device batch size of 8 for training and 4 for evaluation, gradient accumulation over 8 steps, a warmup ratio of 0.03, weight decay of 0.01 and mixed-precision (FP16) training. Each network flow was converted into a compact JSON record with a fixed-order feature vector, which was then passed to the model through an efficient instruction prompt to perform binary classification (*normal* vs *attack*). Training was conducted without exposing the model to the dataset's unseen *attack* categories, and generalization was assessed by evaluating performance on the corresponding held-out portion.

5) Hybrid Detectors: We designed hybrid detectors by intregating TDS with neural models (DNN, GPT-2, and Qwen3), combining their scores. We refined neural probabilities with Platt-style scaling [34], while using temperature scaling to TDS, SBD, and ABD scores to reduce over-confidence and align scales. We computed meta-signals: (i) neural model's confidence, (ii) TDS score, and (iii) their agreement or disagreement. A gating function [35] used these signals to assign greater weight to the neural detector when it was confident and aligned with TDS; otherwise, it shifted more weight to TDS when the neural output was uncertain or contradicted strong anomaly evidence. The fused probability was thresholded with a validation-set to maximize the F1-score.

E. Model Training

We evaluated three datasets: UNSW-NB15, TON-IoT, and CIC-Collection. After preprocessing and the EO-based feature selection, we extracted a subset of flows (hold-out at specific family level) from UNSW-NB15 and CIC-Collection to serve as test samples, removing them from the training sets to ensure that the models never encountered these flows during learning. For the TON-IoT, we randomly selected 20% of *attack* flows (hold-out at sample level) as unseen 'zero-day' traffic. The resulting test sets include zero-day *attacks*, known *attacks*, and *normal* flows, which we later used to evaluate both the overall attack detection and zero-day detection performance. The remaining non-zero-day flows were partitioned into training and validation sets using a stratified 80/20 split on the binary attack label. Each dataset was then assessed using three attack detector categories: (i) TDS (SBD and ABD), (ii) Neural Detectors (DNN, GPT-2, and Qwen3), and (iii) Hybrid Detectors (Hybrid1(TDS-DNN), Hybrid2(TDS-GPT-2), and Hybrid3(TDS-Qwen3)).

IV. Experimental Setup and Evaluation Metrics

This section outlines the training setup, baseline models, and evaluation metrics used to assess the proposed detectors. All models are trained solely on the non–zero-day portion of the training split, with fixed hyperparameters to ensure reproducibility. Traditional baselines include SBD, ABD, and their combined variant (TDS), while the EO-selected DNN, the GPT-2 flow detector, and the Qwen3 detector serve as learned baselines in the same feature space. All hybrid models (i.e. Hybrid1(TDS-DNN), Hybrid2(TDS-GPT-2), and Hybrid3(TDS-Qwen3)) are evaluated under identical train, validation, and test partitions.

A. Experimental Setup

We performed experiments on two platforms: the CAIL-HPC cluster equipped with an NVIDIA GeForce RTX 3080, and Kaggle's cloud environment providing an NVIDIA Tesla P100 GPU. The experimental results reported in the paper were obtained under these hardware settings. All experiments were implemented in Python, using scikit-learn for preprocessing and classical baselines, XGBoost [30] for EO-based fitness evaluation, and PyTorch for neural detectors. Each dataset was converted into a binary classification setup (*normal* vs *attack*), with specific attack families or samples designated as zero-days. The EO-based feature selection was applied to the preprocessed training split—excluding all zero-day flows—using a population of 30 and 50 iterations, and an equilibrium pool of the top four individuals per iteration. The final EO-converged subset was formed as the common input feature space for all detectors.

For neural network detectors, the feedforward DNN using EO-selected features was trained for 30 epochs (batch size 128) with the Adam optimizer (learning rate 10^{-3}) and binary cross-entropy loss. The GPT-2 flow transformer presented each flow as a short feature sequence (length 32, stride 4) and employed a 4-layer, 8-head, 512-dimensional decoder-only

architecture with 0.1 dropout and an MLP classifier. The model was trained for 8 epochs (batch size 32) using AdamW (learning rate 10^{-4}, weight decay 0.01), gradient clipping and binary cross-entropy loss. The Qwen3 detector was fine-tuned with the LoRA-based approach [33]. All hyperparameters were manually tuned on the validation split and fixed across runs.

B. Evaluation Protocol and Metrics

For each detector, the model output was interpreted as an attack probability. Here, SBD, ABD, TDS, DNN, GPT-2 and Qwen3 used a fixed decision threshold of 0.5 as a symmetric decision boundary to avoid model specific bias during testing [2]. For hybrid models, the threshold was selected on the validation split by sweeping the range [0.4, 0.6] and choosing the value maximizing the F1-score; this value was fixed and applied to test splits. To comprehensively assess the performance of the models, we used the following evaluation metrics: Accuracy, Precision, Recall, F1-score, and AUC-ROC [1] and Zero-Day-Recall [19]. It is noted that AUC-ROC captures the threshold-independent discrimination ability, while Zero-Day-Recall measures detection of previously unseen attack families (i.e., never observed during training) [19].

V. Results Analysis

This section presents a systematic performance evaluation of various zero-day detection models across three datasets.

A. Overall Performance Analysis

Table I summarizes a comparative performance of traditional, neural, and hybrid detectors on UNSW-NB15, TON-IoT, and CIC-Collection datasets with zero-day setting with respect to Accuracy, Precision, Recall, and AUC-ROC. Figures 2(a) to (c) highlight the F1-score and Zero-Day-Recall values for all experimented models with all three datasets.

For the UNSW-NB15 dataset, the traditional baselines show limited effectiveness. The SBD detector achieves moderate performance (Accuracy = 0.617, F1-score = 0.69, Zero-Day-Recall = 0.37), whereas the ABD detector performs even less reliably (Accuracy = 0.54, F1-score = 0.57, Zero-Day-Recall = 0.36). On the other hand, TDS(SBD,ABD) detector (with max-fusion) consistently exceeds both components, improving overall Recall and zero-day coverage (F1-score = 0.74, Zero-Day-Recall = 0.57), though it still falls short of the neural detectors. In this case, the DNN detector achieves strong performance (F1-score = 0.90, AUC-ROC = 0.97, Zero-Day-Recall = 0.81), while the GPT-2 flow detector provides a further gain in overall performance and zero-day detection (F1-score = 0.91, Accuracy = 0.89, Zero-Day-Recall = 0.81). The fine-tuned Qwen3-0.6B detector performs poorly as a standalone model, with very low Recall and negligible zero-day coverage. The proposed hybrid detectors offer the best balance: Hybrid1(TDS-DNN) and Hybrid2(TDS-GPT-2) detectors raise Zero-Day-Recall to 0.83 while slightly improving F1-score, while Hybrid3(TDS-Qwen3) detector exhibits reduced performance compared to other hybrid detectors.

For the TON-IoT dataset, the traditional detectors achieve moderate performance: the SBD attains an F1-score of 0.49 with a Zero-Day-Recall of 0.34, the ABD improves to an F1-score of 0.67 and Zero-Day-Recall of 0.51, and their fusion, TDS(SBD,ABD), further increases performance to an F1-score of 0.76 and Zero-Day-Recall of 0.63. The DNN and GPT-2 detectors perform almost perfectly on this dataset (achieving F1-score$>= 0.98$ and Zero-Day-Recall≈ 0.99), leaving little room for additional gains from TDS-based hybrid detectors. In contrast, the fine-tuned Qwen3 model again yields the weakest results, with very low F1-score and zero-day coverage. Notably, fusing Qwen3 with TDS in Hybrid3(TDS–Qwen3) raises the F1-score to 0.90 and Zero-Day-Recall to 0.94, demonstrating that even a weak LLM detector can benefit substantially from the score-level integration with a traditional IDS.

For the CIC-Collection dataset, the traditional baselines remain weak: the SBD and ABD detectors achieve F1-scores of 0.20 and 0.38, with Zero-Day-Recall below 0.20. In addition, the TDS(SBD,ABD) offers only a modest gain with F1-score of 0.38 and Zero-Day-Recall of 0.32. The DNN and GPT-2 detectors raise the overall F1-score to approximately 0.66 and 0.63, yet both exhibit extremely low Zero-Day-Recall which are < 0.04. The fine-tuned Qwen3 again produces very poor Recall and negligible zero-day detection. In this setting, the hybrid detectors demonstrate clear benefits: Hybrid1(TDS-DNN) losses some accuracy but with higher Recall and zero-day coverage (approximately 0.90 and 0.51 respectively), while Hybrid2(TDS-GPT-2) and Hybrid3(TDS-Qwen3) detectors achieve better accuracy than Hybrid1(TDS-DNN). Notably, Hybrid3(TDS-Qwen3) attains a Zero-Day-Recall of approximately 0.57, the highest among all detectors.

Overall, the Zero-Day-Recall increases almost monotonically from the traditional detectors to the neural models and finally to the strongest hybrid detectors on UNSW-NB15 and TON-IoT. In contrast, the results on CIC-Collection show a pronounced collapse of all standalone neural and LLM-based detectors under extreme traffic imbalance. These observations suggest that when rare attacks are severely underrepresented, transformer-based IDSs alone may overlook substantial portions of zero-day flows, while fusing them with lightweight traditional detectors highlights the value of TDS-guided fusion in surfacing such rare threats. Although the Qwen3 detector was fine-tuned using LoRA for a fair comparison, it underperforms as a standalone model due to hallucinations and inconsistent label assignment; however, integrating its outputs with TDS detector consistently enhances overall detection performance.

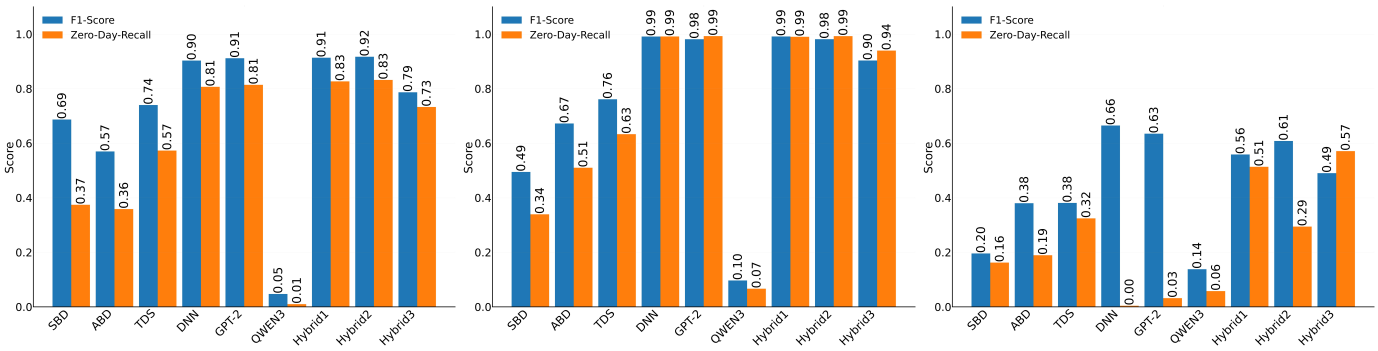
B. Cross-Dataset Zero-Day Behaviour

Across the three datasets, the overall performance of the evaluated models varies substantially, primarily because the definition of zero-day attacks and the underlying data distributions differ across datasets.

With the UNSW-NB15, zero-day evaluation is conducted at the family level: entire zero-day *attack* family (see Section IV.E) are excluded from training and only intro-

TABLE I: Overall Performance of Detection Models on UNSW-NB15, TON-IoT, and CIC-Collection Datasets with respect to Accuracy, Precision, Recall and AUC-ROC.

Detection Models	Network Datasets											
	UNSW-NB15				TON-IoT				CIC-Collection			
	Accuracy	Precision	Recall	AUC-ROC	Accuracy	Precision	Recall	AUC-ROC	Accuracy	Precision	Recall	AUC-ROC
SBD	0.6167	0.7782	0.6146	0.6180	0.4018	0.9406	0.3355	0.5953	0.6854	0.2176	0.1779	0.5012
ABD	0.5382	0.7850	0.4470	0.5912	0.5623	0.9694	0.5145	0.7018	0.6977	0.3402	0.4303	0.6007
TDS(SBD,ABD)	0.6471	0.7459	0.7339	0.5967	0.6528	0.9522	0.6338	0.7081	0.6280	0.4325	0.5442	0.5921
DNN	0.8778	0.9815	0.8324	0.9732	0.9844	0.9907	0.9914	0.9927	0.8470	0.9806	0.5029	0.8563
GPT-2	0.8863	0.9841	0.8556	0.9733	0.9668	0.9707	0.9919	0.9123	0.8072	0.7412	0.5551	0.8527
Qwen3	0.3322	0.7333	0.0243	0.5028	0.1915	0.9351	0.0510	0.5156	0.7187	0.9926	0.0741	0.5369
Hybrid1(TDS-DNN)	0.8892	0.9859	0.8585	0.9745	0.9849	0.9925	0.9902	0.9913	0.6940	0.4051	0.8994	0.9396
Hybrid2(TDS-GPT-2)	0.8924	0.9786	0.8681	0.9748	0.9670	0.9710	0.9918	0.9383	0.8282	0.9762	0.4416	0.7729
Hybrid3(TDS-Qwen3)	0.6865	0.7305	0.8522	0.5972	0.8425	0.9441	0.8657	0.7884	0.7187	0.5739	0.5426	0.6237



(a) UNSW-NB15 (b) TON-IoT (c) CIC-Collection

Fig. 2: F1-score and Zero-Day-Recall performance of detection models on three network datasets. Hybrid1 = Hybrid1(TDS-DNN), Hybrid2 = Hybrid2(TDS-GPT-2), and Hybrid3 = Hybrid3(TDS-Qwen3).

duced during testing. These zero-day flows are produced by tools and behaviours distinct from those observed in the training families, while still residing in the same feature space and traffic context. Under this setting, the traditional TDS baseline achieves only moderate Zero-Day-Recall, and the DNN, GPT-2, and their hybrid variants provide clear—though not saturated—improvements in both F1-score and Zero-Day-Recall. The fine-tuned Qwen3 model performs noticeably worse, whereas Hybrid3(TDS–Qwen3) offers a substantial improvement. Overall, this configuration represents a genuinely challenging semantic zero-day scenario, requiring detectors to generalize to new attack types rather than merely to unseen instances of known attacks.

For the TON-IoT dataset, the original labels distinguish only between *normal* and *attack* flows. Zero-day traffic is defined by randomly holding 20% of the samples as unseen zero-day instances, while training uses the remaining non–zero-day *attack* flows. The resulting splits are heavily attack-dominated (approximately 72% attacks in training and 87% in testing), and the EO-procedure selects a compact subset of 13 highly informative features. Because the held-out zero-day flows originate from the same attack distribution as the training attacks (a sample-level hold-out), the learned decision boundary transfers almost perfectly: both neural detectors and hybrid models achieve accuracy, F1-score, and Zero-Day-Recall nearly 0.99. This configuration thus assesses sample-level generalization rather than family-level evaluation.

Finally, with the CIC-Collection, the rare *Web Attack* traffic is treated as the zero-day family. This yields a substantially more challenging scenario: *Web Attack* flows are extremely scarce (975 out of approximately 9.17M, with only 37 zero-day samples in the test split) and exhibit behaviour distinct from the dominant DoS, Bot, and Brute-Force traffic. Because the CIC-Collection zero-day evaluation contains only 37 held-out Web Attack samples, the reported Zero-Day-Recall should be interpreted cautiously as a high-variance rare-class indicator rather than a stable population-level estimate. Under this severe class and family imbalance, the DNN and GPT-2 detectors still learn strong decision boundaries for the frequent attack types, achieving high overall F1-score, but their Zero-Day-Recall on *Web Attack* remains very low. The fine-tuned Qwen3 also performs poorly, detecting very low number of zero-day attacks. The hybrid fusion improves zero-day coverage (Zero-Day-Recall rising to nearly 0.57), although the extremely small number of *Web Attack* test samples makes the zero-day metric statistically unstable.

In summary, the UNSW-NB15, TON-IoT, and CIC-Collection datasets capture three different zero-day challenges: (i) family-level zero-day under a balanced benchmark (UNSW-NB15), (ii) *sample-level* zero-day drawn from the same *normal/attack* distribution (TON-IoT), and (iii) a *rare, structurally distinct* zero-day family under extreme imbalance (CIC-Collection). Although the EO-based feature selection, transformer models, and hybrid fusion

improve performance across all settings, the achievable Zero-Day-Recall is naturally constrained by the severity of the distribution shift and the rarity of the zero-day class.

VI. Conclusion and Future Work

In this paper, we presented a zero-day attack detection framework that integrates an EO-based feature selection, lightweight traditional attack detectors, and neural network flow models across three heterogeneous datasets. Our approach started with deriving a compact and discriminative feature subset using an EO search guided by cross-validated XGBoost recall and F1-score, while strictly excluding all flows belonging to the held-out zero-day families during training. On this reduced representation, we implemented a percentile-based signature-based detector (SBD), a distance-ratio anomaly-based detector (ABD), and their fusion (TDS), alongside a deep neural network (DNN), a GPT-2 flow transformer, and a fine-tuned Qwen3 model. Finally, we proposed hybrid detectors that combine neural model outputs with TDS scores at the probability level, enabling more robust zero-day detection across diverse traffic conditions.

The empirical evaluation under a strict zero-day protocol revealed a clear performance stratification across model families. Overall, the progression from TDS to DNN, GPT-2, Qwen3, and ultimately the hybrid variants demonstrates that an EO-based feature selection, combined with the fusion of neural and traditional detector scores, achieved the most robust configuration in this setting. Throughout the experiments, we focused primarily on comparing performance variation across detectors and datasets. Due to space constraints, we did not report the computational overhead and significance testing which will be addressed in the upcoming journal version.

There are several promising directions for future work. We aim to explore principled prompt-design methods tailored to network-flow semantics. We plan to investigate fine-tuning pipelines for LLMs, such as GPT-4 and other LLMs to better align them with intrusion-detection tasks with prompt engineering and hyper tuning. We want to address deployment challenges in resource-constrained networking environments by reducing computational and latency overheads. Incorporating uncertainty estimation may further enhance zero-day detection and support reliable real-time operation. Together, these directions represent important steps toward practical, zero-day-aware intrusion detection in real-world networks.

References

[1] S. Binhulayyil, S. Li, and N. Saxena, "IoT vulnerability detection using featureless LLM CyBert model," in *Proc. IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications*, 2024.

[2] M. F. Al-Hammouri, Y. Otoum, R. Atwa, and A. Nayak, "Hybrid LLM-enhanced intrusion detection for zero-day threats in IoT networks," arXiv:2507.07413, 2025.

[3] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "Zero-day attack detection: A systematic literature review," *Artif. Intell. Rev.*, 2023.

[4] V. Babaey and H. R. Faragardi, "Detecting zero-day web attacks with an ensemble of LSTM, GRU, and stacked autoencoders," *Computers*, 2025.

[5] Y. Otoum and A. Nayak, "AS-IDS: Anomaly and signature based IDS for the IoT," *J. Netw. Syst. Manage.*, 2021.

[6] A. Mirza, S. Arshad, M. H. Yousaf, and M. A. Azam, "ZDBERTa: Zero-day cyberattack detection in IOV," *Computers*, 2025.

[7] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on ML algorithms and model performance," *Technologies*, 2021.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017.

[9] Y. Otoum, A. Asad, and A. Nayak, "LLM-based threat detection and prevention framework for IoT ecosystems," arXiv:2505.00240, 2025.

[10] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, and N. Tihanyi, "Generative AI and LLM for cyber security," SSRN 4853709, 2024.

[11] Y. Li, Z. Xiang, N. D. Bastian, D. Song, and B. Li, "IDS-agent: An LLM agent for explainable intrusion detection in IoT networks," 2024.

[12] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, 2020.

[13] M. N. Swileh and S. Zhang, "Unseen attack detection in SDN using BERT," *AI*, 2025.

[14] Z. Zhang, W. Yin, and X. Li, "A novel GPT-based framework for anomaly detection in system logs," arXiv:2510.16044, 2025.

[15] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," arXiv:2505.09388, 2025.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network IDS," in *Proc. MILCOM & MilCIS*, 2015.

[17] A. Bhowmik, "Ton_iot network dataset," 2025. [Online]. Available: https://www.kaggle.com/dsv/12369350

[18] L. D'hooge, "CIC-Collection," 2022. [Online]. Available: https://www.kaggle.com/dsv/4477547

[19] M. A. Alsuwaiket, "ZeroDay-LLM: A LLM for zero-day threat detection," *Information*, 2025.

[20] P. P. Ioulianou, V. G. Vassilakis, I. D. Moscholios, and M. D. Logothetis, "A signature-based IDS for IoT," *IEICE Proceedings Series*, vol. 32, no. SESSION02_3, 2018.

[21] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of IDS: Techniques, datasets and challenges," *Cybersecurity*, 2019.

[22] R.-H. Hwang, M.-C. Peng, C.-W. Huang, P.-C. Lin, and V.-L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, 2020.

[23] M. Sato, H. Yamaki, and H. Takakura, "Unknown attacks detection using feature extraction from anomaly-based IDS alerts," in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet*, 2012.

[24] Y. Yang and X. Peng, "BERT-based network for intrusion detection system," *EURASIP J. Inf. Security*, 2025.

[25] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, 2020.

[26] R. Parmar and N. Bhatt, "A comprehensive review intrusion detection in-vehicle networks," *Emerging Perspectives and Applications of Computational Intelligence and Smart Systems*, 2026.

[27] L. Dhanabal and S. P. Shantharajah, "NSL-KDD dataset for IDS classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, 2015.

[28] E. A. Nack, M. C. McKenzie, and N. D. Bastian, "ACI-IoT-2023: Dataset for IoT network security analysis," in *Proc. IEEE MILCOM*, 2024.

[29] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and for large-scale attacks in IoT," *Sensors*, 2023.

[30] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, 2018.

[31] R. Kumar and D. Sharma, "Signature-anomaly intrusion detection algorithm," in *Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, 2018.

[32] A. AlEroud and G. Karabatis, "A contextual anomaly detection to discover zero-day attacks," in *Proc. Int. Conf. Cyber Security*, 2012.

[33] A. Chavan, Z. Liu, D. Gupta, E. P. Xing, and Z. Shen, "LoRA for parameter-efficient fine-tuning," arXiv:2306.07967, 2023.

[34] N. Phelps, D. J. Lizotte, and D. G. Woolford, "Using Platt's scaling for calibration," arXiv:2410.18144, 2024.

[35] M. Su and M. Basu, "Gating improves neural network performance," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, 2001, pp. 2159–2164.