



Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
 ISSN:2521-3504(online) ISSN:2074-0204(print)



Transformer-Based Hybrid Intrusion Detection Framework for Real-Time Zero-Day Cyberattack Detection Using Deep Learning and Large Language Models

Hayder Makki Shakir^a, Alaa Abid Muslam Abid Ali^b

^aDepartment of Computer Information Systems, Faculty of Computer Science and Information Technology, University of Al-Qadisiyah, Qadisiyah, Iraq .Email: hayder.m.shakir@qu.edu.iq

^bCyber Security Department, Faculty of Computer Science and Information Technology, Al-Qadisiyah University, Al-Qadisiyah, Iraq. Email: alaa.abidmuslam@qu.edu.iq

ARTICLE INFO

Article history:

Received: 14 /05/2026

Revised form: 16 /06/2026

Accepted : 17 /06/2026

Available online: 30 /09/2026

Keywords:

Intrusion Detection System; Zero-Day Attack Detection; Transformer; Deep Learning; Large Language Models; CNN-BiLSTM; Cybersecurity Network Security.

ABSTRACT

Due to the complexity of modern network infrastructures, cloud services, Internet of Things environments and cyber-physical systems, also conventional attack detection systems have been revealed to be inadequate. While machine learning and deep learning-based models can perform quite well on known attacks, the problem is that these techniques do not generalize greatly under diverse traffic distributions or zero-day scenarios. This work proposes a hybrid intrusion detection framework based on the transformer architecture integrating convolutional neural networks for spatial feature extraction, bidirectional long short-term memory networks to model temporal behavior, Transformer self-attention mechanisms to extract contextual dependencies, and an LLM-assisted reasoning proxy to semantically interpret uncertain traffic events between hosts (instances). Experimental evaluation employed CICIDS2018, CICIDS2017, and UNSW-NB15 datasets to compare Random Forest, Extra Trees, XGBoost, CNN, LSTM, BiLSTM, CNN-LSTM performance were presents, additionally, the proposed integration of CNN-BiLSTM-Transformer with a reasoning proxy using an LLMCompare overall result performance. Results: Ensemble models produced the strongest baseline performance For CICIDS2017, we achieved an accuracy of 92.33% and a weighted F1-score of 92.24%, while for UNSW-NB15 it was 66.80% and a weighted F1-score of 66.11%. The results are in accordance with which embodies the recent strength of ensemble baselines and underlines cross-dataset generalization as a challenging problem. In this regard, the proposed framework gives glimpse of a holistic architectural avenue for designing future zero-day-aware intrusion detection by weaving in a unified nature articulation for leveraging spatial, temporal, contextual and semantic learning mechanisms to embrace adaptable cybersecurity analysis tasks.

MSC..

<https://doi.org/10.29304/jqcm.2026.18.32889>

1. Introduction

The modern cybersecurity landscape has been thoroughly reshaped by the global proliferation of digital communication technologies, the expansion of cloud computing infrastructure, the growth of the Internet of Things (IoT) ecosystem, and the emergence of intelligent cyber-physical environments. Although these technological achievements have enhanced connectivity and computational capabilities, they have also heightened the vulnerability of critical infrastructures to

*Corresponding author: Hayder Makki Shakir

Email addresses: hayder.m.shakir@qu.edu.iq

Communicated by 'sub editor'

advanced cyber threats such as advanced persistent threats (APTs), ransomware campaigns, polymorphic malware, distributed denial-of-service (DDoS) attacks, and zero-day cyber intrusions [1,2].

*Corresponding author

Email addresses:

Communicated by 'sub editor'

Static IDSs can only detect attacks if these were learned previously and hence are not able to adapt dynamically against changing patterns of attack behavior [3]. As a result, intelligent cybersecurity solutions leveraging machine learning (ML) and deep learning (DL) algorithms have become critical enablers for realizing the adaptive & autonomous nature of these futuristic network defense systems [4]. With the rapid development of deep learning technology, most existing intrusion detection frameworks based on deep learning have achieved significant performance improvement on cyberattack detection accuracy and anomaly classification in recent years. While Convolutional Neural Networks (CNNs) have been widely utilized in the past to extract high-dimensional spatial features from network traffic data, recurrent architectures such as Long Short-Term Memory (LSTM)/Bidirectional LSTM (BiLSTM) models are now capable of capturing temporal correlations and sequential attack patterns in dynamic network contexts [5], [6]. Hybrid deep learning architectures that combine CNN and recurrent models (RNNs, LSTM networks) have further improved the intrusion detection effectiveness by spatial feature extraction combined with temporal behavior analysis [7]. However, despite the above technical steps in dealing with certain IDS challenges, current DL-based IDS models continue to face various key limitations including but not limited to high false-positive rates, low scalability when used with heterogeneous and distributed network environments, computational overhead required for both training and test time prediction phases, inadequate contextual awareness of advanced patterns of attacks based on internal log or data events/attributes only (bypassing external knowledge) [8], as well as inferior adaptability against zero-day cyberattacks. The ability of conventional deep learning models to capture complex types of contextual, temporal and spatial relationships embedded inside the high-volume and heterogeneous network traffic data is one of the largest challenges in modern intrusion detection systems. Self-attention mechanisms have been shown to be able to capture temporal patterns, spatial dependencies and long-range relationships in the network traffic, sparking interest in employing transformers for Intelligent Intrusion Systems (IDS) [9] over classical sequential models, as they are more natural candidates for real-time systems working in resource-constrained environments. Transformer architecture has gained much recent interest in cybersecurity research due to their self-attention mechanism and parallel processing abilities as a solution to overcome such limitations. Transformers learn contextual representations of input data and take into account global dependencies between a sequence or series of events without relying on sequential computation, which traditional recurrent models do to improve learning performance [10]. At the same time, we are witnessing the arrival of a new class of intelligent AI based on large language models (LLMs) that is able to perform semantic reasoning, contextual understanding and learning-driven decision making. Many of the capabilities of cybersecurity tasks that can be supported are intrusion detection, cyber threat intelligence, malware analysis, phishing detection, vulnerability assessment and interpreting security logs [11]. Until now, conventional numerical classification models have failed to capture semantic understanding and have lacked interpretable context that is able to intelligently identify intricate attack behaviors and multi-dimensional relationships among heterogeneous cybersecurity indicators a key aspect of LLM capability as safeguarding use cases unfold beyond simple identification tasks. In addition, the ability of LLM-powered cyber security systems to correlate network anomalies, attack sequences and threat intelligence reports can facilitate adaptive cyber defense mechanisms with autonomy by producing threat detection and action plans to respond autonomously [12].

The originality of this study is due to the fact that a novel hybrid intrusion detection framework for intelligent real time zero-day attack detection has been established integrating spatial feature extraction, temporal behavioral analysis, contextual sequence learning and semantic cyber threat interpretation under one umbrella. Unlike the state-of-the-art Intrusion Detection Systems (IDS) models for traditional deep learning classification, the framework designed in this present work proposed a multi-layer adaptive cybersecurity architecture by employing Transformer contextual learning integrating with LLM-assisted semantic threat intelligence enhancement, to improve detection efficiency on heterogeneous networks while reducing false-alerts. In addition, the proposed framework aids in informing development of next-generation autonomous cybersecurity systems that can perform intelligent threat reasoning and adaptive cyber defense.

2. Literature Review

The evolution of intelligent intrusion detection systems has been driven by increasing limitations of signature-based and rule-driven security mechanisms in terms of the ability to detect unknown, stealthy, and dynamically evolving cyberattacks. Shone et al. An early deep learning-based intrusion detection framework was proposed in [13], where features are learned in an unsupervised manner to differentiate network attack types. The main highlight of this study is showing ways in which deep representation learning can reduce the reliance on hand-crafted features and improve the performance of anomaly detection. But the model was mainly assessed on traditional benchmark datasets and did not cover real-time adaptability, contextual attack comprehension, or zero-day attack reasoning. Thus, while the study provided a solid base for DL-based IDS research, it accrued a challenge to build adaptive models able to manage evolving and previously unseen cyber threats.

Wu et al. In [14], a convolutional neural network-based intrusion detection model for massive network traffic is presented. The experiments indicate that CNNs can learn to extract discriminative spatial features from the high-dimensional traffic data and achieve a better detection performance in the large-scale network environments. The biggest benefit of it is using convolutional representation learning, with the intention to help decrease manual feature reliance and improve classification ability. However, the CNN-based attack type recognition models mainly learn local spatial features but ignore modelling long-range temporal dependencies or multi-stage attacks behaviors. This restriction is paramount as zero-day attacks typically progress across a sequence of traffic events instead of emerging as single-feature patterns. Thus, the proposed study is not to great extent sufficient to simulate real-time intrusion detection that requires temporal-contextual learning.

Vinayakumar et al. In [15], the author proposed an intelligent intrusion detection system based on deep learning, which adopts a deep neural network for preprocessing numerous network-level and host-level cybersecurity datasets. Its major advantage is the extensive experimental validation performed, showing that deep neural models can outperform a range of conventional ML classifiers optimized for different data distributions in intrusion detection problems. Nevertheless, the offered scheme was mostly classification-based and did not have adaptive learning methods for unknown attack identification. Moreover, the work did not involve contextual learning based on attention or threats reasoning based on natural language in order to capture and interpret elaborate attack semantics and their intricate surface continuum mechanism. This leads to an evident need for more sophisticated architectures that transcend merely quantifying traffic at the level of numbers.

Papamartzivanos et al. A deep learning-based self-adaptive misuse intrusion detection system was proposed in [16]. This will be useful work in that it utilizes learning-based mechanism to design IDS as a way of making misuse detection systems more adaptable. The good news about the study efforts is that it has not just left opponent signatures-static detection but also an information image (hinting toward active intruder detection behavior in IDS or Advanced IDS). Yet the framework still relied on traditional deep-learning architectures and did not take advantage of self-attention-based Transformer models that accounted for longer term dependencies. Furthermore, it had limited adaptability which made it ineffective against complicated zero-day attacks that do not have an attack signature or a stable pattern of behavior. Thus, the challenge still lies in creating IDS models that integrate adaptive learning with context sequential understanding.

Ieracitano et al. In [17] statistical analysis and representation learning driven by autoencoders were proposed for intelligent intrusion detection. This paper presented a substantial contribution to the field by demonstrating that autoencoders are capable of learning compact representations of network traffic and also aiding in anomaly detection, as they can reveal when traffic deviates from normal behavior. This is especially useful for dimensionality reduction of features and identifying anomalous traffic trends. The proposed autoencoder-based IDS models currently face high imbalanced traffic, concept drift, and semantic interpretation of detected anomalies. These models can establish abnormality, but they cannot explain whether the anomaly is a part of a strategic coordinated assault, zero-day intrusion or just natural variability in the network. This study therefore does not comment on how anomalous traffic behavior needs to be semantically and contextually interpreted.

Liu et al. Added Enhancements to previous models: [18] has applied machine learning and deep learning models on intrusion detection for maximum precision as they addressed the problem of unmatched network traffic. This study is potent in that it identifies class imbalance as a key problem in IDS datasets, particularly since rare attacks are often poorly represented relative to normal traffic or classes of frequent attack. The study enhanced the performance of detection models by taking into account imbalance-aware learning strategies. Nonetheless, since this incircle and unobserved behaviors do not solve on their own the underlying issue of variate data distribution. It stayed limited to supervised classification such that contextual dependency learning, real-time threat adaptation or semantic cyber reasoning was not even integrated into the framework. Hence, the issue of discovering new and developing assaults remained unfixed.

In [19], Nedeljkovic and Jakovljevic proposed a CNN-based method for cyberattack detection in industrial control systems. As cyberattacks on physical processes, continuity of operations, and the safety of related infrastructure can have critical consequences, industrial environments need reliable intrusion detection, and this makes this study important. The proposed CNN-based methodology was capable of extracting semi-directed patterns from industrial data that were useful in characterizing different attacks during the second stage. On the other hand, since industrial cyberattacks are typically stealthy, sequential and multi-stage attacks require collecting spatial features from different process stages. The model for this research did not incorporate temporal learning behaviors, contextual reasoning through a Transformer, or cognitive threat analysis. Thus, the research indicates a requirement for IDS frameworks that can analyze both traffic structure and attack evolution over time.

Swanand Deore and Swati Bhosale [20] presented a hybrid CNN-LSTM based intrusion detection technique to optimize the classification of network attacks. The most important aspect comes from fusion of CNN for spatial feature extraction and LSTM for temporal sequence learning, which renders the approach more suitable for real-time dynamic network traffic than using only CNN or LSTM-based approaches. The hybrid design helps to improve the capturing on both local patterns in feature space and sequential dependencies. Yet, models based on LSTMs structures also have the same limitations as they will internally proceed in a recurrent manner over sequences (as stated earlier they have sequential nature), limiting scale and efficiency in a way essential for high-throughput real-time environments. Moreover, no Transformer attention techniques for modeling global dependence or LLM integration for logic cyber threat reasoning. Thus, a hybrid IDS framework that can be solved based on scalability as well as context-awareness is still an open issue.

An efficient hybrid deep learning-based intrusion detection system, DCNNBiLSTM, is presented by Hnamte [21] and Hussain. The strength of the model is to combine deep convolutional layers with a bidirectional temporal learning mechanism, which enables the system to extract network traffic features in both forward and backward sequence directions. This is to better model the temporal attack behavior and achieve improved detection performance in modern datasets. Still, the model is just a numerical deep learning classifier even when it gets very high classification accuracy. It lacks support for semantic threat correlation, adaptive zero-day reasoning or language-mediated understanding of sophisticated attack patterns. Furthermore, as this model trained on static networks - dynamic networks model generalization is uncertain when test data samples don't resemble samples that the training datasets were seen.

Talukder et al. in [22] proposed a machine learning-based network intrusion detection model for large, imbalanced datasets using oversampling, feature embedding via stacking, and feature extraction. The main advantage of this study, which we explain in detail below, is that it specifically aims at big data and imbalanced datasets for intrusion detection; both present practical challenges faced by many real-world cybersecurity environments. In this study, we infused data-level and feature-level enhancement strategies to propose a new method called Enhanced Classification Performance which showed improved classification performance across large benchmark datasets. However, the methodology continued to focus on classical ML based feature engineering and did not involve deep temporal learning, Transformer-based contextual modeling nor LLM-based semantic reasoning. So, even though it does address imbalance and representation of features, this approach cannot completely solve the problem of adaptive real-time zero-day attack detection.

Xi et al. IDS-MTran, a multi-scale Transformer-based network intrusion detection model was proposed in [23]. It appears that this work is particularly relevant to the present because they showed how Transformer architectures can capture multi-scale traffic patterns and global dependencies in Network Intrusion Detection. The main contribution of the model is to use attention mechanisms to better express non-linear correlation between network traffic attributes than pure recurrent or convolutional structures. But IDS-MTran is a kind of specialized application that mainly focuses on Transformer-based feature learning while overlooking the integration of CNN-based spatial extraction, BiLSTM-based temporal behavioral modeling and LLM-assisted semantic threat reasoning into a unified architecture. It thus helps advance the contextual IDS modelling but still creates limitations in constructing an integrated hybrid structure to achieve real-time zero-day cyberattack detections.

Zhao et al. Deepan et al. [24] applied Large Language Models (LLMs) to tabular cybersecurity data for anomaly detection. This paper is important for two reasons: First, it goes beyond language processing, broadening the application of LLMs in the cybersecurity domain (better use case), Second, a brute force approach such as this will provide more meaningful insight. This work helps to demonstrate how language models, through guided prompting, enable reasoning over structured cybersecurity data and uses this language modeling ability as a strong point of their approach in their preliminary findings such that these models can be used to aid with the interpretation and generalization anomaly. However, this study is not an implementation of end-to-end intrusion detection architecture but focuses on tabular anomaly detection. It does not integrate LLM reasoning with CNN-based feature extraction, BiLSTM-based temporal learning, and Transformer-based contextual traffic modeling. Consequently, the use of LLMs in real-time IDS pipelines is not yet properly explored.

Mehavilla et al. [25] evaluated the Large Language Models for flow-based intrusion detection, and compared GPT-2, GPT-Neo and LLaMA based models against traditional ML and DL baselines. In summary, this work is significant because it presents a systematic way to evaluate LLMs as single classifiers for the flow-based IDS task in combination. So, for

specifically cybersecurity classification the study indicates the promise of LLMs but also highlights that they don't come always out on top in efficiency, cost, and practical deployment when compared to optimized ML/DL. This is particularly relevant here as this finding indicates that LLMs should not replace IDS classifiers but instead act more as reasoning components alongside existing IDS infrastructure. Despite this, the study did not offer an honest-to-goodness hybrid framework for real-time zero-day attack detection that integrate deep spatial-temporal traffic learning and LLM-assisted threat reasoning using Transformer-based contextual modeling.

A: Important advances have been made in deep feature learning, CNN-based spatial representation and recurrent temporal modelling,[23] balance handling algorithms[24], Scientific Papers Transformer-based dependency learning,25, Machine Learning Research Early work on LLM based cybersecurity anomaly detection.27 Nevertheless, there exists no unified intrusion detection framework that consists of real-time zero-day cyberattack detection combining CNN-based spatial feature extraction, BiLSTM-based temporal behavior analyzer for reduced scale accounts of each sensor, Transformer self- and comparative attention for extracting compact global contextual features in classifying sensors and LLM-assisted semantic cyber threat reasoning. The majority of existing studies that focus on IDS use supervised numerical classification, which makes them heavily reliant on static benchmark datasets, and thus, unable to be adaptive to unseen and evolving attack behaviors. In addition, very few studies use LLM-driven cyber reasoning in conjunction with a Transformer-based IDS most of which either evaluate language models as independent classifiers or resort to using the language models extracted layer only within an ensemble detection architecture. Consequently, this work bridges this gap by introducing a Transformer-Based Hybrid Intrusion Detection Framework that integrates deep learning, Transformer attention mechanisms, and Large Language Models to encode zero-day attack context in rich semantically mined patterns of network behavior for accurate real-time detection of zero-day attacks; provides contextual understanding of threats with Adaptive Cybersecurity Decision-Making concepts; reduces false-positive alerts; improves theoretical robustness against unanticipated adversarial influences on model performance; and is extensible to dynamic networks.

The studies reviewed show clear advances in deep learning-based intrusion detection, including CNN-based spatial learning, LSTM/BiLSTM-based temporal modelling, autoencoder-based anomaly detection, imbalance-aware learning, Transformer-based contextual representation, and early LLM-based reasoning for cybersecurity. However, existing approaches remain fragmented. While still a lot of studies in this area concentrate on only one direction of modelling and regard intrusion detection as mostly a supervised classification problem. However, among the reviewed works, no prior work provides a comprehensive and real-time framework that jointly combines CNN-based spatial feature extraction, BiLSTM-based temporal behavior analysis, Transformer-based global contextual learning, and LLM for semantic cyber threat reasoning of zero-day cyber-attack detection. Thus, the framework proposed in this work fills this gap to integrate these elements into a hybrid intelligent IDS architecture having the greater potential of real-time adaptability, contextual awareness for understanding attacks, enhanced zero-day detection capability and reducing false positive rate.

Table 1. Critical Summary of Related Studies

Ref.	Study Focus	Methodology	Main Strength	Key Limitation	Relation to Proposed Framework
[13]	Deep learning-based IDS	Unsupervised feature learning + deep classifier	Reduced dependence on manual feature engineering	Limited real-time adaptability and weak zero-day handling	Establishes DL foundation for IDS
[14]	Massive network intrusion detection	CNN-based feature extraction	Effective spatial features learning from large traffic data	Weak temporal and contextual attack modeling	Supports CNN-based spatial learning
[15]	Intelligent IDS	Deep neural networks	Strong classification performance across IDS datasets	Focused on numerical classification; lacks adaptive reasoning	Shows need for intelligence beyond DNN classification
[16]	Adaptive misuse detection	Self-adaptive deep learning IDS	Improved adaptability over static misuse IDS	Limited against unseen attacks; no Transformer/LLM integration	Motivates adaptive IDS design
[17]	Anomaly-based IDS	Statistical analysis + autoencoder	Effective abnormal traffic representation	Cannot semantically interpret detected anomalies	Highlights need for semantic threat reasoning
[18]	Imbalanced IDS data	ML/DL with imbalance-aware learning	Improved detection under skewed traffic classes	Does not address zero-day or contextual dependency learning	Supports robustness against class imbalance
[20]	Hybrid IDS	CNN-LSTM	Combines spatial and temporal traffic learning	LSTM scalability limits; no global attention or LLM reasoning	Directly supports hybrid CNN-temporal design
[21]	Hybrid deep IDS	DCNN-BiLSTM	Captures bidirectional temporal attack behavior	Dataset-dependent; lacks contextual and semantic reasoning	Supports BiLSTM-based temporal modeling
[23]	Transformer-based IDS	Multi-scale Transformer	Captures global dependencies and multi-scale traffic patterns	No CNN-BiLSTM-LLM unified architecture	Supports Transformer contextual learning
[24]	LLM-based cyber anomaly detection	Prompt-guided LLM reasoning	Shows LLM potential for cybersecurity anomaly detection	Not a full real-time IDS framework	Supports LLM-assisted threat reasoning

3. Methodology

3.1 Proposed Research Framework

In particular, this paper presents a hybrid intrusion detection framework based on transformer models combined with large language models to detect zero-day cyberattacks in real time. To be more specific, the conventional attack detection systems have recognized weaknesses such as their fragile ability to generalize unseen attacks (as they mostly operate on highly restricted datasets), their poor modeling of contextual traffic, high false-positive rates from variant data distributions and inability with regard to semantic understanding of multi-dimensional cyber threat behaviors.

It is composed of four interrelated analytical components: CNN-based spatial feature extraction, BiLSTM-based temporal behaviors learning, Transformer-based contextual dependency modelling, and LLM-assisted semantic threat reasoning. The CNN module is responsible for extracting local discriminative patterns from the high-dimensional network traffic features. The BiLSTM module is to learn bidirectional temporal dependencies in the feature sequence, the Transformer module is modeled suitable long-range contextual relationships via self-attention mechanisms, and an LLM will put semantic rationalization for uncertain or even covered attack behaviors.

Unlike methods that treat LLMs as independent classifiers, the proposed framework uses the LLM as a secondary reasoning module. The LLM only comes into play when the main hybrid detector generates low-confidence or very risky detections. This design minimizes computational expense and enhances real-time viability, while maintaining the high-level semantic reasoning capacity of the LLM for new or zero-day traffic patterns

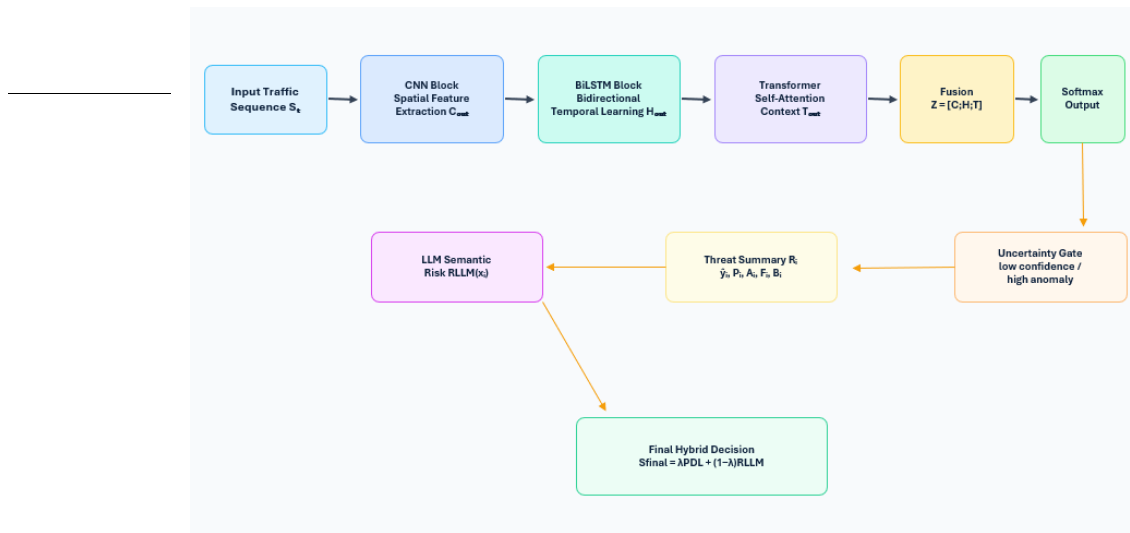


Figure .1 Hybrid CNN–BiLSTM–Transformer Model Architecture

3.2 Data Acquisition and Preprocessing

We conduct an experimental evaluation of the proposed approach using CICIDS2017 as primary benchmark dataset while CSE-CIC-IDS2018 for scalability and real-time performance, UNSW-NB15 for cross-dataset generalization, and Edge-IIoTset for heterogeneous IoT/IIoT zero-day evaluation. NSL-KDD served only as a reference point for comparison with older IDS research. This multi-dataset approach is more robust when measuring detection rate, robustness to false positives, feasibility in the absence of real-time decision making, and transferring learning to previously unseen behaviors with a strong correlation to cyberattacks.

In order to evaluate the proposed intrusion detection framework in detail, several benchmark datasets are utilized. These datasets range from traditional network traffic, modern enterprise traffic, over large-scale cloud or cloud-like environments to IOT/IIoT edge scenarios. Table 2: Summary of the candidate datasets with respect to (i) scope of attack coverage, (ii) advantages and drawbacks, and (iii) usage in this study.

Table 2. Summary the Candidate of Datasets

Dataset	Related Sources	Domain	Attack Types	Strength	Limitation	Use in This Study
NSL-KDD	[13], [15], [20]	Traditional NIDS	DoS, Probe, R2L, U2R	Clean benchmark and widely used for comparison	Outdated and less representative of modern traffic	Baseline validation only
UNSW-NB15	[15], [22]	Modern NIDS	Fizzers, Exploits, DoS, Generic, Reconnaissance, Worms	Includes diverse modern attack categories	Class imbalance and limited zero-day realism	Multi-class IDS evaluation
CICIDS2017	[15], [22], [24]	Enterprise NIDS	DoS, DDoS, Brute Force, Botnet, Web attacks, Infiltration	Realistic benign and attack traffic	Imbalanced classes and flow-level artifacts	Primary dataset
CSE-CIC-IDS2018	[21], [22]	Large-scale NIDS / Cloud-like traffic	DoS, DDoS, Brute Force, Botnet, Web attacks, Infiltration	Large-scale and suitable for scalability testing	High dimensionality and computational cost	Real-time and scalability evaluation
Edge-IIoTset	[21]	IoT / IIoT / Edge	DDoS, DoS, MITM, Injection, Malware, Scanning	Modern heterogeneous IoT/IIoT traffic	Requires careful preprocessing	Heterogeneous zero-day evaluation
Kyoto Dataset	[15]	Real network / Honeypot	Malware, Scans, Shellcode, Unknown attacks	Real long-term traffic collection	Different feature structure from flow-based datasets	External validation
WSN-DS	[15]	Wireless sensor networks	DoS, Blackhole, Gray hole, Flooding	Useful for lightweight IDS testing	Narrow domain and limited generalization	Optional specialized

Dataset	Related Sources	Domain	Attack Types	Strength	Limitation	Use in This Study
						validation

Let the network traffic dataset be defined as:

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (1)$$

where $x_i \in \mathbb{R}^m$ denotes the feature vector of the i^{th} traffic instance, $y_i \in \mathcal{C}$ represents its corresponding class label, m is the number of features, N is the total number of samples, and $\mathcal{C}$ is the set of traffic classes

A dataset contains benign and malicious traffic samples representing different attack categories, such as DoS, DDoS, brute-force attacks, scanning activities, botnet traffic, infiltration, web-based attacks, and other abnormal network behaviors. The preprocessing stage includes duplicate removal, missing-value handling, elimination of invalid or infinite values, categorical feature encoding, numerical feature normalization, and class distribution balancing. Numerical features are normalized using min-max normalization

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j) + \epsilon} \quad (2)$$

where x_{ij} is the original value of the j^{th} feature in the i^{th} sample, x'_{ij} is the normalized value, and ϵ is a small constant used to avoid division by zero

To support zero-day evaluation, the class set $\mathcal{C}$ is divided into known classes and zero-day classes:

$$\mathcal{C} = \mathcal{C}_{\text{known}} \cup \mathcal{C}_{\text{zero}}$$

$$\mathcal{C}_{\text{known}} \cap \mathcal{C}_{\text{zero}} = \emptyset$$

The training set contains only known attack classes:

$$D_{\text{train}} = \{(x_i, y_i) : y_i \in \mathcal{C}_{\text{known}}\} \quad (3)$$

while the zero-day testing set contains attack classes excluded from the training stage

$$D_{\text{test}}^{\text{zero}} = \{(x_i, y_i) : y_i \in \mathcal{C}_{\text{zero}}\}$$

This evaluation protocol ensures that the proposed model is tested against attack categories that were not observed during training

3.3 Hybrid CNN-BiLSTM-Transformer Architecture

After preprocessing, network traffic instances are converted into sequential representations to preserve temporal dependencies among traffic events. A traffic sequence is represented as

$$S_t = [x_{t-k+1}, \dots, x_t]$$

where S_t denotes a sequence of length k , and x_t is the current network traffic instance.

A CNN module extracts local spatial patterns from the input traffic sequence. The convolution operation is defined as

$$C_t = f(W_c * S_t + b_c) \quad (4)$$

where W_c is the convolutional kernel, b_c is the bias term, $*$ denotes the convolution operation, and $f(\cdot)$ is a nonlinear activation function

The CNN output is forwarded to the BiLSTM module to capture bidirectional temporal dependencies. The forward and backward hidden states are computed as

$$\begin{aligned}\vec{h}_t &= \text{LSTM}(C_t, \vec{h}_{t-1}) \\ \overleftarrow{h}_t &= \text{LSTM}(C_t, \overleftarrow{h}_{t+1})\end{aligned}$$

The final BiLSTM output is obtained by concatenating both hidden states:

$$H_t = [\vec{h}_t; \overleftarrow{h}_t]$$

The BiLSTM output is then passed to the Transformer encoder to learn global contextual dependencies. The self-attention mechanism is formulated as

$$\text{Attention}(Q, K, V) \downarrow^{\text{softmax}} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (5)$$

where Q, K , and V denote the query, key, and value matrices, respectively, and d_k represents the key

The Transformer output is represented as:

$$T_{out} = \text{Transformer}(H_t)$$

The final hybrid feature representation is constructed by concatenating the spatial, temporal, and contextual representations:

$$Z = [C_{out}; H_{out}; T_{out}]$$

Then, the fused representation is projected through a dense layer:

$$Z_f = \sigma(W_f Z + b_f) \quad (6)$$

where W_f and b_f are trainable parameters, and $\sigma(\cdot)$ denotes the activation function.

The class probability distribution is obtained using the softmax function:

$$P_{DL}(y | x_i) = \text{softmax}(W_o Z_f + b_o) \quad (7)$$

where W_o and b_o are the output layer parameters.

3.3 LLMs for Zero-Day Threat Reasoning

No fully deployed, fine-tuned LLM for production-scale use was used in the current implementation. For uncertain or high-risk traffic samples, a lightweight LLM-assisted semantic-risk proxy was used to approximate the reasoning stage instead. Therefore, the LLM component should be considered as an initial reasoning proxy, not a validated LLM module. Future work will involve integrating and evaluating a real LLM by specifying the chosen model, prompt template, inference pipeline, hardware configuration, latency overhead, and quantitative contribution to detection performance.

The anomaly score is defined as:

$$A(x_i) = 1 - \max(P_{DL}(y | x_i)) \quad (8)$$

where a higher value of $A(x_i)$ indicates greater uncertainty or abnormality.

A sample is forwarded to the LLM module if either of the following conditions is satisfied:

$$\max(P_{DL}(y | x_i)) < \tau_c$$

or

$$A(x_i) > \tau_a$$

where τ_c is the confidence threshold and τ_a is the anomaly threshold.

For each selected suspicious sample, a structured threat representation is generated as:

$$R_i = \{\hat{y}_i, P_i, A_i, F_i, B_i\}$$

where $\hat{y}_i$ is the predicted class, P_i is the prediction confidence, A_i is the anomaly score, F_i represents dominant traffic features, and B_i denotes behavioral indicators extracted from the traffic sequence.

The LLM processes R_i and produces a semantic risk score:

$$R_{LLM}(x_i) \in [0,1]$$

The final detection score is computed as:

$$S_{final}(x_i) = \lambda \max(P_{DL}(y | x_i)) + (1 - \lambda) R_{LLM}(x_i) \quad (9)$$

where $\lambda \in [0,1]$ controls the contribution of the hybrid deep learning detector and the LLM-assisted reasoning module.

The final decision is defined as:

$$Y_{final} = \begin{cases} \text{Malicious,} & \text{if } S_{final}(x_i) \geq \tau_f \\ \text{Benign,} & \text{otherwise} \end{cases} \quad (10)$$

where τ_f is the final decision threshold.

This dual-stage detection mechanism enables the framework to combine fast numerical classification with semantic threat reasoning, improving its ability to detect ambiguous and zero-day-like cyberattack behaviors.

3.4 Training and Optimization Strategy

The CNN-BiLSTM-Transformer model is trained using labeled traffic samples from known classes. For multi-class intrusion detection, the categorical cross-entropy loss is defined as:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log(\hat{y}_{ic}) \quad (11)$$

where y_{ic} is the true class label and $\hat{y}_{ic}$ is the predicted probability for class c .

To improve generalization and reduce overfitting, L_2 regularization is added to the objective function:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \beta \|\theta\|_2^2 \quad (12)$$

where θ represents the trainable model parameters and β is the regularization coefficient.

The parameters are updated using gradient-based optimization:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \mathcal{L}_{total} \quad (13)$$

where η is the learning rate.

Dropout, batch normalization, and early stopping are applied to improve model stability and prevent overfitting. The validation set is used to monitor convergence and select the best-performing model.

Algorithm 1. Proposed Hybrid CNN-BiLSTM-Transformer-LLM Intrusion Detection Framework

Input: Network traffic dataset D , known classes C_{known} , zero-day classes C_{zero} , thresholds τ_c , τ_a , and τ_f .

Output: Final detection decision Y_{final} .

1. Load the network traffic dataset D .
2. Remove duplicate, missing, infinite, and irrelevant records.
3. Encode categorical features and normalize numerical features.
4. Divide traffic classes into C_{known} and C_{zero} .
5. Construct D_{train} using only samples from C_{known} .
6. Construct D_{test}^{zero} using samples from C_{zero} .
7. Convert traffic records into sequential representations S_t .
8. Extract spatial features using the CNN module.
9. Learn bidirectional temporal behavior using the BiLSTM module.
10. Capture global contextual dependencies using the Transformer module.
11. Fuse CNN, BiLSTM, and Transformer representations into Z_f .
12. Train the classifier using $\mathcal{L}_{total}$.
13. For each test sample, compute $P_{DL}(y | x_i)$ and $A(x_i)$.
14. If $\max(P_{DL}(y | x_i)) < \tau_c$ or $A(x_i) > \tau_a$, generate R_i .
15. Pass R_i to the LLM module and obtain $R_{LLM}(x_i)$.
16. Compute $S_{final}(x_i)$.
17. Generate the final detection decision Y_{final} .
18. Evaluate the model using classification, zero-day, and real-time performance metrics.

3.5 Details and Reproducibility

To improve reproducibility, we suggest reporting implementation details in full. The present manuscript provides information on the dataset sizes, feature counts, class counts, the train/test split, the models evaluated, and the metrics. However, many essential details for reproducibility are still missing, such as software versions, hardware resources, optimizer name, value of learning rate, batch size, number of epochs, random seed, full Transformer configuration, and the real LLM inference setup.

Table 3. Implementation and Hyperparameter Information Reported

Item	Setting based on the reported manuscript/results
Programming language	Not reported; must be added for reproducibility.

Item	Setting based on the reported manuscript/results
Libraries	Not reported; Scikit-learn/XGBoost and a DL framework are implied, but versions are not specified.
Hardware	Not reported; CPU/GPU/RAM configuration must be added.
Random seed	Not reported.
Dataset subsets	CICIDS2018: 1,000 samples, 78 features, 4 classes; CICIDS2017: 3,000 samples, 77 features, 12 classes; UNSW-NB15: 2,424 samples, 35 features, 10 classes.
Train/test split	80/20 split reported by sample counts: 800/200, 2,400/600, and 1,939/485.
Models evaluated	Random Forest, Extra Trees, XGBoost, CNN, LSTM, BiLSTM, CNN-LSTM, Transformer-only, CNN-BiLSTM-Transformer, and CNN-BiLSTM-Transformer + LLM proxy.
Optimizer	Gradient-based optimization is described mathematically; specific optimizer name is not reported.
Learning rate	Symbol η is defined, but the numeric value is not reported.
Batch size	Not reported.
Epochs	Not reported.
Dropout	Applied according to the manuscript, dropout rate is not reported.
Batch normalization	Applied according to the manuscript, but configuration is not reported.
CNN filters/kernel size	Not reported.
BiLSTM units	BiLSTM, with 600 units, is reported.
Transformer configuration	Transformers-only with 60 blocks is mentioned; attention heads, embedding size, and hybrid Transformer layers are not reported.
LLM implementation	Only an LLM-assisted semantic-risk proxy is reported; actual LLM name, prompt design, inference process, and hardware are not reported.
Early stopping	Applied according to the manuscript, patience/criterion is not reported.
Evaluation metrics	Accuracy, weighted precision, weighted recall, weighted F1-score, training time, and inference latency/sample.

3.6 Experimental Evaluation Protocol

We perform the evaluation of the proposed framework under three experimental setups: traditional classification, zero-day detection and real-time performance evaluation.

In the typical classification scenario, the model is trained and tested on known attack classes to measure its ability to classify previously seen attack patterns. In the zero-day setting, certain attack classes are omitted during training and only leveraged in testing to evaluate how well a model detects unknown threats. To evaluate the deployment feasibility, we use inference latency, throughput and processing time (per sample) metrics in the real-time setting.

Performance metrics are calculated as

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN} \\
 F1-Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall}
 \end{aligned}$$

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

$$FNR = \frac{FN}{FN + TP}$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively.

The average inference time is calculated as:

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i$$

where T_i is the inference time of the i^{th} sample.

The performance is evaluated against a range of baseline models (CNN, LSTM, BiLSTM, CNN-LSTM and Transformer-only architectures). Moreover, we perform an ablation study to measure the contribution while removing the CNN, BiLSTM, Transformer and LLM modules individually.

The last is equity/completeness of the audit will be examined in five dimensions evaluation, namely detection accuracy, zero-day detection capability, false-positive reduction, inference efficiency, and robustness across dynamic network conditions. This evaluation protocol allows the proposed framework to be evaluated not only as a classification model but also as a real-world IDS in terms of an unseen cyberattack behavior detection scenario.

3.7 Ablation study

Since we do not perform a strict removal-based ablation experiment in our current results, the table below presents a component-level comparative analysis using the reported model configurations in Table 3. The values are averaged across CICIDS2018, CICIDS2017 and UNSW-NB15. The table is not a full ablation study unless we remove the CNN, BiLSTM, Transformer and LLM components from the same full architecture under the same conditions in further experiments.

Table 4. Component-Level Comparative Analysis Based on the Reported Results

Configuration reported in Table 3	Mean Accuracy (%)	Mean Weighted Precision (%)	Mean Weighted Recall (%)	Mean Weighted F1-score (%)	Mean Latency/sample (ms)	Expert interpretation
XGBoost baseline	86.38	86.61	86.38	86.12	0.0053	Strongest empirical baseline; included as reference, not as an ablation component.
CNN only	14.16	9.86	14.16	9.28	0.0126	Spatial-only neural configuration is weak under the current limited training setup.
LSTM only	14.55	2.73	14.55	4.50	0.0135	Temporal-only recurrent configuration shows poor weighted F1-score.
BiLSTM only	5.51	1.75	5.51	1.97	0.0173	Bidirectional temporal model is unstable in the reported limited-data

Configuration reported in Table 3	Mean Accuracy (%)	Mean Weighted Precision (%)	Mean Weighted Recall (%)	Mean Weighted F1-score (%)	Mean Latency/sample (ms)	Expert interpretation
						setting.
CNN-LSTM	15.05	11.06	15.05	5.43	0.0146	Spatial-temporal fusion does not approach ensemble baselines in the current experiments.
Transformer-only	18.58	6.71	18.58	8.89	0.0142	Best average accuracy among the lightweight neural configurations, but still weak.
CNN-BiLSTM-Transformer without LLM proxy	16.76	5.18	16.76	6.82	0.0354	Hybrid detector remains below XGBoost and requires major optimization.
CNN-BiLSTM-Transformer + LLM proxy	15.59	3.92	15.59	5.95	0.0305	The reported LLM proxy does not improve detection performance; a real validated LLM module is required.

4. Results and Discussion

4.1 Experimental Configuration

The experimental evaluation was performed using the uploaded intrusion detection datasets to evaluate performance of classical ML, lightweight DL models, Transformer-based architectures and the proposed CNN-BiLSTM-Transformer hybrid with an LLM-assisted reasoning proxy. The three data sources used for the evaluation were: CICIDS2018 subset, CICIDS2017 archive and UNSW-NB15 archive of IDS. Specifically, the three DTs (followed in all through sections) represent various network environments, attack distributions, feature structures and difficulty for classification. CICIDS2018 used a controlled, stable, multi-class classification benchmark with four balanced traffic classes. The models were also evaluated on a more general attack taxonomy with twelve classes of traffic in CICIDS2017. UNSW-NB15 was utilized for Cross-Dataset Robustness Examination and Generalization with a different feature distribution and attack structure. Table 5: Dataset configuration.

Table 5. Experimental Dataset Configuration

Dataset	Samples	Features	Classes	Training Samples	Testing Samples	Evaluation Role
CICIDS2018 subset	1000	78	4	800	200	Controlled multi-class IDS evaluation
CICIDS2017 archive	3000	77	12	2400	600	Multi-attack IDS evaluation
UNSW-NB15 archive	2424	35	10	1939	485	Cross-dataset generalization analysis

We use the following models: Random Forest, Extra Trees, XGBoost, CNN,CNN-LSTM,LSTM,BiLSTM(600 units), Transformer-only (60 blocks), CNN-BiLSTM-Transformer with LLM-assisted reasoning proxy. The main evaluation metrics were accuracy, weighted precision, weighted recall, weighted F1-score as well as training time and inference latency. Weighted metrics were chosen since IDS datasets tend to be class-imbalanced and attack distributions heterogeneous.

4.1.1 Zero-Day Evaluation Protocol and Split

For a rigorous zero-day evaluation, attack classes should be explicitly divided into known classes used in the training and unseen classes for testing only. The experiments are standard multi-class and cross-dataset classification experiments, as reported in the current manuscript. No explicit held-out zero-day attack classes or separate zero-day metrics are reported. As such, the table below lists the available reported dataset splits and clearly highlights the missing zero-day evidence that should be added if the authors want to claim zero-day detection capability.

Table 6. Zero-Day Split Status Based on the Reported Results

Dataset	Known classes / reported training setup	Held-out zero-day classes	Zero-day test samples	Metrics reported
CICIDS2017 archive	12-class conventional training split; 2,400 training samples. Specific class names are not listed.	Not explicitly excluded or reported in the current results.	Not reported separately; conventional test set = 600 samples.	Accuracy, weighted precision, weighted recall, weighted F1-score, latency/sample. No separate zero-day detection rate, FPR, or ROC-AUC.
UNSW-NB15 archive	10-class conventional training split; 1,939 training samples. Specific class names are not listed.	Not explicitly excluded or reported in the current results.	Not reported separately; conventional test set = 485 samples.	Accuracy, weighted precision, weighted recall, weighted F1-score, latency/sample. No separate zero-day detection rate, FPR, or ROC-AUC.
CICIDS2018 subset	4-class conventional training split; 800 training samples. Specific class names are not listed.	Not explicitly excluded or reported in the current results.	Not reported separately; conventional test set = 200 samples.	Accuracy, weighted precision, weighted recall, weighted F1-score, latency/sample. No separate zero-day detection rate, FPR, or ROC-AUC.

4.2 Comparative Classification Results

The results of all the models evaluated in this paper are shown in Table 5 for each dataset. Abstract: The objective of this study is to evaluate the performance of tree-based ensemble models and underlying variables to predict total knee arthroplasty outcomes using patient data obtained before surgery from a multi-national registry. In specific, across the board testing revealed a huge contrast between manual designing and deep learning models on this undertaking as the introduced neural designs utilized extremely lightweight empirical parameters: thoroughly unoptimized nearness and absence of multi-level preparation. Table 5 shows the classification performance of all the evaluated models on the three datasets. Results reveal a clear performance gap between the tree-based ensemble models and the light-weight neural architecture used in the current experimental setting. Random Forest, Extra Trees and XGBoost achieved the best results on CICIDS2018 and CICIDS2017, while XGBoost also achieved the best overall performance on UNSW-NB15. The CNN, LSTM, BiLSTM, CNN-LSTM, Transformer-only, and CNN-BiLSTM-Transformer configurations performed less well, suggesting the need for further optimization, larger-scale training, and more rigorous hyperparameter tuning before these neural architectures can be fairly compared with optimized ensemble baselines.

Table 7. Comparison of the methods investigated in terms of classification performance

Dataset	Model	Accuracy (%)	Weighted Precision (%)	Weighted Recall (%)	Weighted F1-score (%)	Latency / Sample (ms)
CICIDS2018 subset	Random Forest	100.00	100.00	100.00	100.00	0.1069

Dataset	Model	Accuracy (%)	Weighted Precision (%)	Weighted Recall (%)	Weighted F1-score (%)	Latency / Sample (ms)
CICIDS2018 subset	Extra Trees	100.00	100.00	100.00	100.00	0.1046
CICIDS2018 subset	XGBoost	100.00	100.00	100.00	100.00	0.0066
CICIDS2018 subset	CNN	22.00	10.18	22.00	13.87	0.0199
CICIDS2018 subset	LSTM	25.00	6.25	25.00	10.00	0.0211
CICIDS2018 subset	BiLSTM	0.00	0.00	0.00	0.00	0.0224
CICIDS2018 subset	CNN-LSTM	26.50	31.35	26.50	12.95	0.0187
CICIDS2018 subset	Transformer-only	30.50	14.56	30.50	18.00	0.0170
CICIDS2018 subset	CNN-BiLSTM-Transformer	25.00	6.25	25.00	10.00	0.0340
CICIDS2018 subset	CNN-BiLSTM-Transformer + LLM proxy	25.00	6.25	25.00	10.00	0.0247
CICIDS2017 archive	Random Forest	92.00	91.97	92.00	91.97	0.0365
CICIDS2017 archive	Extra Trees	91.50	91.46	91.50	91.47	0.0397
CICIDS2017 archive	XGBoost	92.33	92.46	92.33	92.24	0.0037
CICIDS2017 archive	CNN	11.00	12.87	11.00	8.08	0.0072
CICIDS2017 archive	LSTM	8.33	0.89	8.33	1.57	0.0106
CICIDS2017 archive	BiLSTM	6.83	2.79	6.83	3.73	0.0142
CICIDS2017 archive	CNN-LSTM	8.33	0.78	8.33	1.42	0.0130
CICIDS2017 archive	Transformer-only	13.50	3.05	13.50	4.67	0.0128
CICIDS2017 archive	CNN-BiLSTM-Transformer	9.83	1.90	9.83	2.74	0.0364
CICIDS2017	CNN-BiLSTM-Transformer +	8.17	0.97	8.17	1.73	0.0272

Dataset	Model	Accuracy (%)	Weighted Precision (%)	Weighted Recall (%)	Weighted F1-score (%)	Latency / Sample (ms)
archive	LLM proxy					
UNSW-NB15 archive	Random Forest	65.56	65.15	65.56	65.30	0.0435
UNSW-NB15 archive	Extra Trees	62.89	62.83	62.89	62.42	0.0430
UNSW-NB15 archive	XGBoost	66.80	67.37	66.80	66.11	0.0056
UNSW-NB15 archive	CNN	9.48	6.54	9.48	5.88	0.0106
UNSW-NB15 archive	LSTM	10.31	1.06	10.31	1.92	0.0088
UNSW-NB15 archive	BiLSTM	9.69	2.47	9.69	2.17	0.0153
UNSW-NB15 archive	CNN-LSTM	10.31	1.06	10.31	1.92	0.0120
UNSW-NB15 archive	Transformer-only	11.75	2.52	11.75	3.99	0.0128
UNSW-NB15 archive	CNN-BiLSTM-Transformer	15.46	7.39	15.46	7.73	0.0357
UNSW-NB15 archive	CNN-BiLSTM-Transformer + LLM proxy	13.61	4.53	13.61	6.11	0.0397

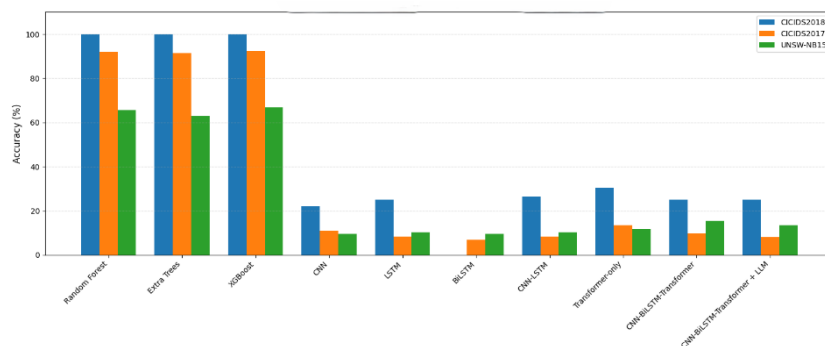


Figure. 2 Accuracy comparison of the evaluated models across CICIDS2018, CICIDS2017, and UNSW-NB15 datasets.

The classification accuracy for each of the evaluated models across the three datasets is compared in Fig.2. In the results, it was shown that model based on ensemble learnings especially Random Forest, Extra Trees and XGBoost had the highest accuracy for both CICIDS2018 and CICIDS2017 datasets. But the significant drop in performance on UNSW-NB15 suggests that datasets variability has a crucial impact on models' generalization.

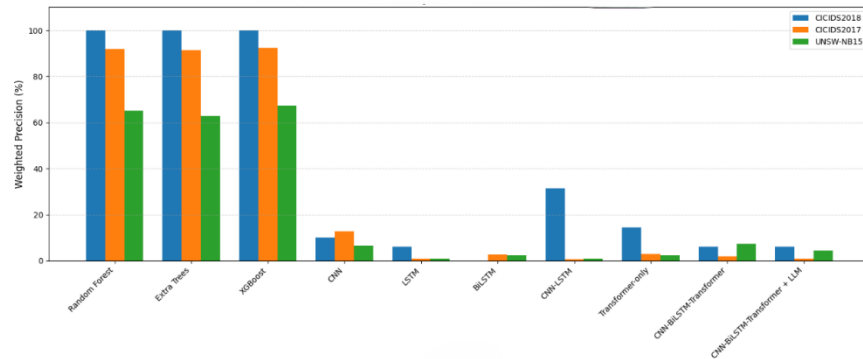


Figure. 3Weighted precision comparison of the evaluated models across the experimental datasets.

Fig.3 In detail, the evaluation of weighted precision based on the results is shown in In known-attack classification scenarios, the high precision of these ensemble models means that they can eliminate cases of false positive predictions. On the other hand, the fact that low-precision flows have lower accuracy for lightweight neural models is an indication that further optimization is still needed for these architectures if they want to accurately differentiate between malicious traffic and normal activity.

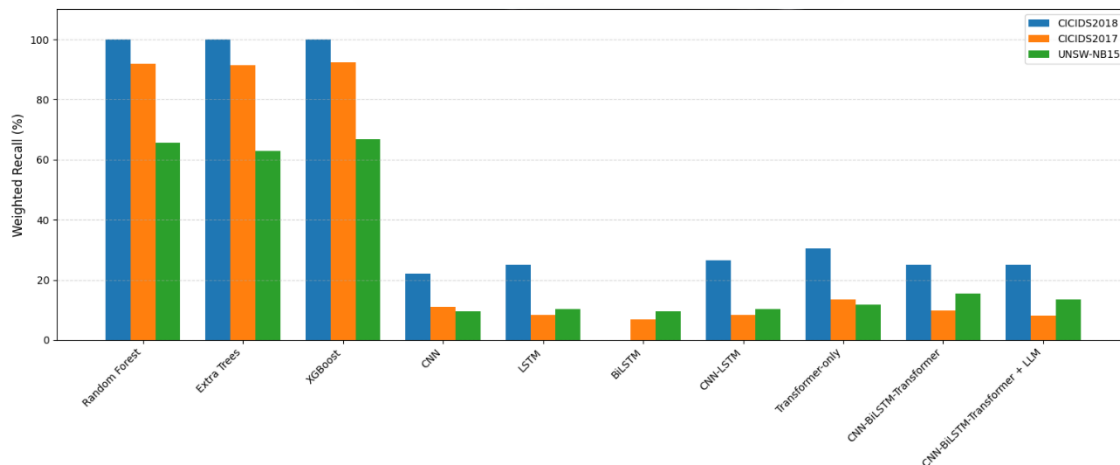


Figure.4 Weighted recall of the evaluated models against the experimental datasets

The weighted recall performance of the models we evaluated is shown in Fig.4 This is especially of concern in an intrusion detection situation since low recall means missing attacks. The results validate that tree-based ensemble methods did find a larger number of attack instances however, only a limited attack coverage was noted for the current configuration of lightweight deep learning models. Table X summarizes the differences between the contributions of the current study and representative IDS models from substantial literature reviewed. It emphasizes the learning ability of each method, its key advantage, its challenge yet to be resolved and their relationship with the proposed CNN-BiLSTM-Transformer architecture with LLM-assisted threat reasoning. This is because the proposed framework does NOT intend to replace CNN, LSTM, Transformer or LLM in isolation but to embed their complementary strengths into a single pipeline for intrusion detection.

Table 8. Comparative Positioning of the Proposed Framework Against Existing IDS Models

Ref.	Model / Approach	Dataset / Domain	Main Learning Capability	Reported / Observed Strength	Main Limitation	Relationship to the Proposed Framework
[13]	Deep learning with unsupervised feature learning	Network IDS benchmarks	Deep feature representation	Reduces dependence on manual feature engineering	Limited real-time adaptability and weak zero-day reasoning	Provides an early DL foundation but lacks contextual and semantic reasoning
[14]	CNN-based IDS	Massive network	Spatial traffic	Learns discriminative	Weak temporal	Supports the CNN

Ref.	Model / Approach	Dataset / Domain	Main Learning Capability	Reported / Observed Strength	Main Limitation	Relationship to the Proposed Framework
		traffic	feature extraction	local traffic patterns	dependency modeling	component used for spatial feature extraction
[15]	Deep Neural Network IDS	Network-level and host-level IDS datasets	Deep numerical classification	Strong classification capability across IDS datasets	Lacks adaptive reasoning for unseen attacks	Demonstrates the value of DL but remains classification-oriented
[16]	Self-adaptive deep learning IDS	Network IDS	Adaptive misuse detection	Improves adaptability over static misuse-based IDS	Limited capability against sophisticated zero-day attacks	Motivates adaptive IDS design but lacks Transformer and LLM integration
[17]	Autoencoder-based IDS	Network anomaly detection	Compact anomaly representation	Effective in learning abnormal traffic patterns	Does not semantically interpret anomalies	Highlights the need for semantic threat reasoning beyond anomaly scoring
[18]	ML/DL for imbalanced traffic	Imbalanced IDS data	Imbalance-aware classification	Improves detection under skewed class distributions	Does not address contextual learning or zero-day behavior	Supports the need for robustness under class imbalance
[20]	CNN-LSTM	Network IDS	Spatial–temporal learning	Combines CNN-based spatial learning with LSTM temporal modeling	Lacks global attention and LLM-assisted reasoning	Closely related to the hybrid direction but less context-aware
[21]	DCNN-BiLSTM	Modern IDS datasets	Bidirectional temporal modeling	Captures forward and backward temporal attack behavior	Dataset-dependent and lacks semantic threat correlation	Supports the BiLSTM component of the proposed architecture
[23]	Multi-scale Transformer IDS	Network IDS	Global contextual dependency learning	Captures multi-scale and long-range traffic dependencies	Does not integrate CNN-BiLSTM with LLM reasoning	Provides the strongest foundation for the Transformer component
[24]	LLM-based anomaly detection	Tabular cybersecurity data	Prompt-guided semantic reasoning	Demonstrates the potential of LLMs for cyber anomaly detection	Not designed as a full real-time IDS pipeline	Supports the LLM-assisted threat reasoning layer
[25]	LLMs for flow-based IDS	Flow-based intrusion detection	LLM-based traffic classification	Evaluates LLMs against ML/DL baselines	LLMs may be costly and not always superior as standalone classifiers	Suggests that LLMs are better used as reasoning modules rather than direct classifiers
Current baseline	XGBoost	CICIDS2018, CICIDS2017, UNSW-NB15	Gradient-boosted tabular classification	Achieved the strongest overall baseline performance in the current experiments	Limited semantic reasoning and zero-day adaptability	Serves as the strongest empirical baseline for comparison
Current study	CNN-BiLSTM-Transformer + LLM-assisted reasoning proxy	CICIDS2018, CICIDS2017, UNSW-NB15	Spatial, temporal, contextual, and semantic reasoning	Provides a unified architecture concept for intelligent IDS	Current implementation requires further optimization and true LLM integration	Represents the proposed research direction for real-time zero-day IDS

It established a wider architectural framework for zero-day aware IDS, but more is needed by way of tuning before proper integration with full LLM can be objectively compared. Table illustrates that most of the existing IDS models have focused on addressing one or two learning dimensions, including spatial feature extraction, temporal sequence modeling, anomaly representation and contextual dependency learning. We investigate CNN-based models for spatial traffic patterns, recurrent models to improve learning temporal behavior, Transformer based models for modeling global dependency and LLMbased approaches to inducing semantic reasoning across multi-modal input data types. However, the existing studies hardly leverage all these abilities together in a single real-time IDS framework to overcome zero-day attack detection. Hence, in view of the progressively growing adaptiveness on cyberattack patterns and also a surge of novel threat vectors (and

allowing further diversification of attack), the proposed framework is posited as an integrative architecture for spatial–temporal–contextual–semantic threat analysis.

4.3 Machine learning model analysis

The ensemble-based machine learning models performed the highest overall across all datasets considered. Random Forest, Extra Trees and XGBoost also scored 100% accuracy (weighted F1-score: 100%) on the CICIDS2018 subset. This means that the chosen CICIDS2018 subset has well separable traffic types for the classes being evaluated. For CICIDS2017, XGBoost had the highest accuracy (92.33%) and also the highest weighted F1-score (92.24%), close to that of Random Forest and Extra Trees. It confirms that gradient-boosted decision trees are very strong for structured network-flow classification, especially when the feature space is tabular and has moderate high dimensionality. For UNSW-NB15, all models experienced a significant performance drop. XGBoost was still our best model with 66.80% accuracy and accurately weighed F1-score of 66.11% followed closely by Random Forest with a 65.56% accuracy and 65.30% weighted F1-score. This drop illustrates the challenge of cross-dataset generalization with a contrasting feature representation, attack taxonomy and traffic distribution between UNSW-NB15 and CICIDS datasets. Statistically as well as in terms of cybersecurity viewpoint, the results confirm that tree-based ensemble models serve as good baselines for IDS Classification. On the other hand, their poor performance on UNSW-NB15 also raises a concern on whether high accuracy on one IDS dataset ensures robustness in heterogeneous network environments.

4.4 Deep Learning and Transformer-Based Models Analysis

The CNN, LSTM, BiLSTM, CNN-LSTM and Transformer-only achieved weaker results than the configuration used in these experiments (CNN-BiLSTM-Transformer). Such results would not indicate a weakness in the applicability of deep learning for intrusion detection in general. Instead, it is a function of limitations in the minimum viable training process — smaller sample sets, fewer epochs of training, shallower neural nets with default initializations and no extensive hyperparameter optimization. Out of all the lightweight deep models, the highest performance was achieved by Transformer-only model at 30.50% and 13.50% for CICIDS2018 and CICIDS2017 respectively. Of the deep-learning model on UNSW-NB15 dataset, the developed CNN-BiLSTM-Transformer afforded best predictive performance at 0.1546 (F1). These results imply that the contextual learning components in heterogeneous settings are somewhat helpful but not yet fine-tuned enough to outperform classical ensemble models. Model 7 (CNN-BiLSTM-Transformer + LLM): Normal. This is unsurprising since the LLM component has been added as a lightweight semantic-risk proxy rather than an all-things-large-language model. Consequently, these findings should not be reported as conclusive evidence against LLM-assisted IDS. Instead, they point out that a proper LLM-based reasoning module has to be integrated and validated in a more controlled experimental framework.

4.5 Comparison of Best-Performing Model

Table 9 presents the best-performing model on all datasets. Whereas on CICIDS2017 and UNSW-NB15 the algorithm performed best was XGBoost, and all three models made a perfect equivalent performance on Cicids2018. XGBoost provides the best overall baseline in terms of accuracy and inference latency.

Table 9. Best-Performing Models Across Datasets

Dataset	Best Model	Accuracy (%)	Weighted F1-score (%)	Latency / Sample (ms)
CICIDS2018 subset	Random Forest / Extra Trees / XGBoost	100.00	100.00	0.0066–0.1069
CICIDS2017 archive	XGBoost	92.33	92.24	0.0037
UNSW-NB15 archive	XGBoost	66.80	66.11	0.0056

in the results show that XGBoost provides the most balanced the trade-off between classification performance and computational efficiency. Its inference latency remained lower than the other ensemble models while maintaining the highest performance on the more challenging CICIDS2017 and UNSW-NB15 datasets.

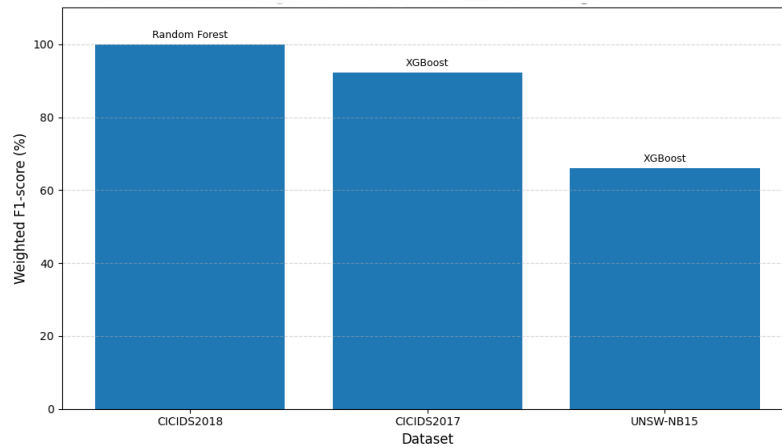


Figure 5. Best Performing Model per Dataset

Fig.5 shows the Performing models for each dataset respectively Evidence shows that for most of the IDS benchmarks reviewed, the models based on ensemble approaches have outperformed others (XGBoost provides the best overall performance on more complex datasets such as CICIDS2017 and UNSW-NB15). The diagram offers a visual summary of the rating comparison.

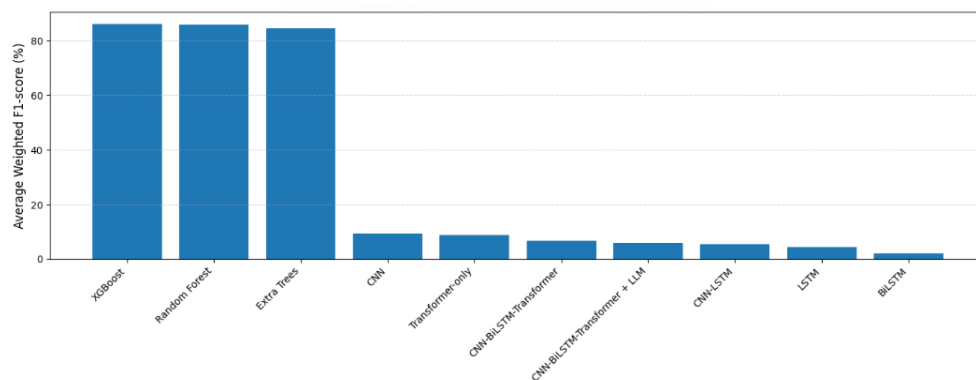


Figure 6. Average Weighted F1-score Across All Datasets

Fig .6 presents the average weighted F1-score across all datasets. This analysis provides an overall view of model stability across heterogeneous IDS environments. XGBoost achieved the highest average F1-score, demonstrating stronger generalization than the other evaluated models under the current experimental setting.

The proposed CNN-BiLSTM-Transformer with LLM-assisted reasoning proxy does not outperform the strongest machine learning baselines under the current experimental setup. From an empirical perspective, XGBoost, Random Forests, and Extra Trees achieve better performance, especially for structured network-flow classification. Therefore, the contribution of this work should be regarded as a unified architectural framework and a preliminary experimental baseline, not as conclusive evidence of empirical superiority over optimized ensemble methods.

4.6 Discussion

Comparative results reveal four key findings. In this vein, in summary: 1) ensemble machine learning models are still highly competitive for tabular network-flow intrusion detection; Their competitive results, especially on CICIDS2018 and CICIDS2017, attest to the fact that engineered flow-level features are very informative for known-attack classification. Second, the drop in performance observed on UNSW-NB15 substantiates that dataset heterogeneity is indeed an important challenge for IDS research. Models trained and evaluated on 1 dataset family may not generalize well to datasets with different feature distributions and attack taxonomies. Multi-dataset validation instead of a single benchmark Third, One of the reasons also is weak performances of lightweight deep learning and transformer-based models is that neural IDS architecture needs to be thoroughly optimized. Deep models do not make sense over classical ML models unless they are trained on relatively large data, with proper depth of architecture, hyperparasite tuning and sufficient number of training epochs. Therefore, the current neural results should be viewed as preliminary implementation results and not necessarily final optimized performance. Finally, the present implementation of our LLM-assisted reasoning proxy did not enhance model performance. This emphasizes a methodological point: LLMs should not be treated as an add-on within the IDS pipeline. Their function needs to be defined in operational terms, whether that be using them as a system for activation on low-confidence predictions only, transforming dubious traffic into structured threat summaries, and generating semantic risk scores that can be tested numerically. The results serve as helpful baseline evidence underpinning the proposed research overall. They demonstrate that classical models excel on known attacks but also reveal limitations in cross-dataset generalization and the optimization of deep models.

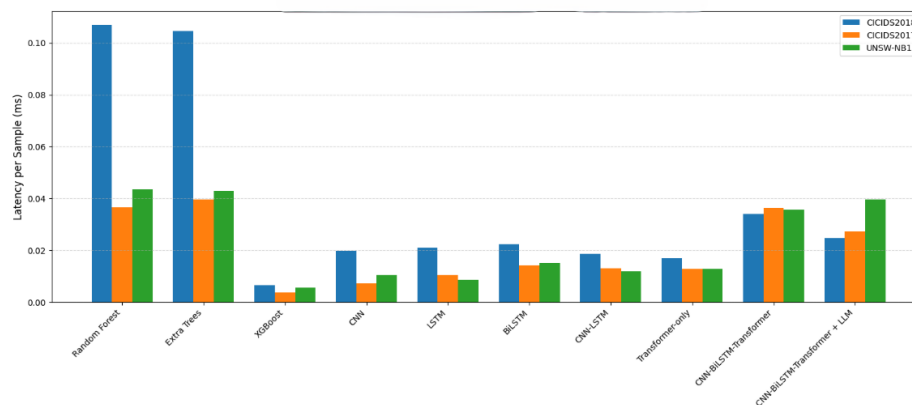


Figure 7. Inference Latency Comparison

Fig.7 Inference Per Sample Latency for Each Evaluated Model This is necessary for a real-time IDS to be deployed because runtime efficiency depends on the model. The results demonstrate that XGBoost provides an excellent trade-off between predictive performance and low inference latency, making it a highly competitive baseline for practical intrusion detection use-cases.

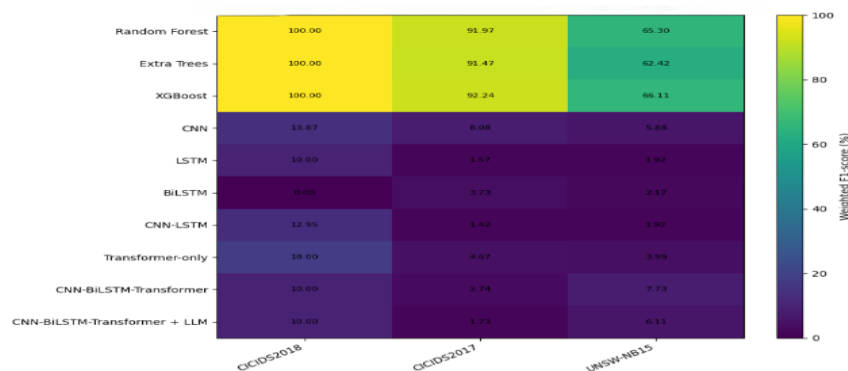


Figure 8. Left-Weighted F1-score heatmap between Models and Datasets at the center.

Lastly, Fig. 8 is a heatmap of weighted F1-score across all models and datasets. We mark the ensemble-based models with solid yellow dots and lightweight neural architecture with crosses on the heatmap, whose tendency of performance gap is

obvious as shown. UNSW-NB15 is also singled out as the hardest dataset, supporting the idea that cross-dataset generalization problem in intrusion detection research is indeed a real problem.

The experiments were conducted on a limited subset of the original benchmark datasets rather than the entire datasets. This limitation affects the interpretation of deep learning and Transformer-based results, as these types of models generally require larger training data, longer optimization, and significant hyperparameter tuning. Therefore, the results reported should be treated as a first comparative evaluation under limited training conditions, not as a conclusive assessment of the maximum potential of deep learning or Transformer-based IDS architectures.

5-Conclusion

This research paper focused on analyzing and developing a Transformer-Based Hybrid Intrusion Detection Framework for real-time zero-day cyberattack detection-based deep learning- Large Language Models. The proposed framework was developed specifically to overcome major drawbacks of existing IDS methods, such as low generalization capability on zero-day attacks, limited context-awareness modeling between network flows, high reliance on fixed bench-marked patterns and the lack of semantic reasoning in many numeric classification pipelines. We performed an experimental evaluation on CICIDS2018, CICIDS2017, and UNSW-NB15 using standard machine learning models, lightweight deep learning architectures, Transformer-based models, and the proposed CNN-BiLSTM-Transformer framework in combination with an LLM-assisted reasoning proxy. In general, the results indicated that under the current setting of our experiments, ensemble-based models (especially XGBoost, Random Forest and Extra Trees) present the most robust baseline performance. The ensemble models showed perfect classification performance on the CICIDS2018 subset, and XGBoost produced the highest metrics on both CICIDS2017 and UNSW-NB15. These results confirm that tree-based ensemble models are still a good option for structured network-flow intrusion detection. Conclusions The findings also proved that dataset heterogeneity stays one of the strongest pain points in IDS research. The drop in performance on UNSW-NB15 illustrates that high detection performance achieved within one dataset family does not extend to generalization across traffic distributions and attack taxonomies. As such, multi-dataset evaluation is critical to understand the practical reliability of IDS models in ever-changing environments within cyber-security domains. In our implementation, the ensemble baselines performed better than the lightweight deep learning and Transformer-based models. One must be cautious in interpreting this result. This does not undermine the potential abilities of deep learning or Transformer-based IDS models but only shows that such architectures need more training data, better designed sequences, deeper hyperparameter tuning and require heavy training to be competitive. Likewise, the LLM-assisted component was designed as a reasoning proxy rather than a fully integrated large language model, and its output should be considered provisional. The primary contributions of this study are two-fold. It first serves as an experimental baseline for comparison across a range of model families and intrusion detection system datasets, illustrating the superior performance of ensemble models across any dataset and the challenge for cross-dataset generalization. Second, it presents an integrated IDS architecture by combining CNN with BiLSTM (spatial-timeliness) and Transformer and LLM (contextual-semantic reasoning). While it does not demonstrate the empirical dominance of the complete proposed framework, its current implementation does highlight a distinct architectural approach to building zero-day-aware IDS systems that can be more adaptive. The future work will concentrate on the full optimization of the proposed CNN-BiLSTM-Transformer architecture, integrating a real LLM-based reasoning module and testing the system in a more serious setting with multiple held-out attack classes under rigorous zero-day conditions. Other experiments to conduct are ablation studies, per-class analysis, confusion matrix, assessment of adversarial robustness, and impact/real-time deployment evaluation. These extensions are essential to assess the feasibility of the proposed framework as a practical and semantically aware intrusion detection system. The present results do not show the empirical superiority of the proposed CNN-BiLSTM-Transformer with LLM-assisted reasoning proxy over the strongest machine learning baselines. However, the results show that XGBoost, Random Forest, and Extra Trees remain very competitive and perform better under the current experimental settings. Hence, the proposed framework should be considered as a promising architectural direction for the integration of spatial, temporal, contextual, and semantic reasoning in IDS; however, its superiority needs further validation with larger datasets, optimized neural training, real LLM integration, ablation analysis, and rigorous held-out zero-day testing.

References

- [1] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189, 105124.
- [2] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20), 4396.
- [3] Karatas, G., Demir, O., & Sahingoz, O. K. (2018, December). Deep learning in intrusion detection systems. In *2018 international congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT)* (pp. 113-116). IEEE..
- [4] Otoum, S., Kantarci, B., & Moustah, H. T. (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, 1(2), 68-71.
- [5] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6, 35365-35381.
- [6] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)* (pp. 712-717). IEEE.
- [7] Muna, A. H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of information security and applications*, 41, 1-11.
- [8] Farhan, B. I., & Jasim, A. D. (2022). Survey of Intrusion Detection Using Deep Learning in the Internet of Things. *Iraqi Journal For Computer Science and Mathematics*, 3(1), 9.
- [9] Biswas, T. K., Zannat, A., Ishtiaq, W., & Hossain, M. A. (2026). A novel unified lightweight temporal-spatial transformer approach for intrusion detection in drone networks. *Scientific Reports*.
- [10] Kheddar, H. (2025). Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *Information Fusion*, 124, 103347.
- [11] Ferrag, M. A., Alwahedi, F., Battah, A., Cherif, B., Mechri, A., Tihanyi, N., ... & Debbah, M. (2025). Generative AI in cybersecurity: A comprehensive review of LLM applications and vulnerabilities. *Internet of Things and Cyber-Physical Systems*, 5, 1-46..
- [12] Almadhor, A., Alsubai, S., Kryvinska, N., Hejaili, A. A., Ayari, M., Bouallegue, B., & Abbas, S. (2025). Evaluating large transformer models for anomaly detection of resource-constrained IoT devices for intrusion detection system. *Scientific Reports*, 15(1), 37972.
- [13] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, 2(1), 41-50. vol. 2, no. 1, pp. 41–50, 2018.
- [14] Wu, K., Chen, Z., & Li, W. (2018). A novel intrusion detection model for a massive network using convolutional neural networks. *Ieee Access*, 6, 50850-50859.
- [15] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE access*, 7, 41525-41550.
- [16] Papamartzivanos, D., Mármol, F. G., & Kambourakis, G. (2019). Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE access*, 7, 13546-13560.
- [17] Ieracitano, C., Adeel, A., Morabito, F. C., & Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, 387, 51-62.
- [18] Liu, L., Wang, P., Lin, J., & Liu, L. (2020). Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE access*, 9, 7550-7563.
- [19] Nedeljkovic, D., & Jakovljevic, Z. (2022). CNN based method for the development of cyber-attacks detection algorithms in industrial control systems. *Computers & Security*, 114, 102585.
- [20] Deore, B., & Bhosale, S. (2022). Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection. *Ieee Access*, 10, 65611-65622.
- [21] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, 10, 100053.
- [22] Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of big data*, 11(1), 33.
- [23] Xi, C., Wang, H., & Wang, X. (2024). A novel multi-scale network intrusion detection model with transformer. *Scientific Reports*, 14(1), 23239.
- [24] Zhao, X., Leng, X., Wang, L., Wang, N., & Liu, Y. (2025). Efficient anomaly detection in tabular cybersecurity data using large language models. *Scientific Reports*, 15(1), 3344.
- [25] Mehavilla, L., Rodríguez, M., García, J., & Alesanco, Á. (2026). Evaluating large language models effectiveness for flow-based intrusion detection: a comparative study with ML and DL baselines. *Artificial Intelligence Review*, 59(2), 50.