International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.2 (2025) pp. 3465-3485
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Deep Guard: A Novel Transformer-Based Framework for Real-Time Threat Detection in Heterogeneous Cyber Environments

Pradeep K R[1]*, Lakshmi B N[2], M Varaprasad Rao[3], N. Sree Divya[4], M. Sree Vani[5], K.Shailaja[6]

[1*] Department of AI &amp; ML, BMS Institute of Technology and Management, Post Box No.6443, Avalahalli, DB Road, Yelahanka, Bangalore - 560064.
* **Corresponding Author Email:** pradeepkr@bmsit.in - **ORCID:** 0000-0002-9045-801X

[2] Department of Computer Science and Engineering, BMS Institute of Technology and Management Bangalore, Karnataka.
Email: lakshmibn@bmsit.in- **ORCID**: 0000-0002-9216-0348

[3] Department CSE(DS),Professor , CVR College of Engineering , Hyderabad, Telangana, India
Email: varam78@gmail.com – **ORCID:** 0000-0002-9045-801Y

[4] Department: Information Technology , Assistant Professor , Mahatma Gandhi Institute of Technology , Hyderabad, Telangana, India
Email: nsreedivya_it@mgit.ac.in– **ORCID:** 0000 0003 0366 4251

[5] Professor, Department of CSE, BVRIT HYDERABAD COLLEGE OF ENGINEERING, FOR WOMEN, Hyderabad-500090
Email: pmksrct@gmail.com – **ORCID:** 0009-0002-2827-9210

[6] Associate Professor , Department of CSE Vasavi College of Engineering Telangana india.
Email: e.shailaja@staff.vce.ac.in – **ORCID:** 0000-0001-7641-6828

Article Info:

DOI: 10.22399/ijcesen.2394
Received : 21 March 2025
Accepted : 20 May 2025

Keywords :

Deep Learning
Intrusion Detection
Transformer Networks
IoT Security
Anomaly Detection

Abstract:

With evolving cyber threats in Internet of Things (IoT) and Industrial IoT (IIoT) networks, challenges with heterogeneous data and dynamic attack patterns cannot be addressed using traditional intrusion detection systems (IDS). We present DeepGuard, a novel deep learning framework for these challenges. DeepGuard enhances detection in space heterogeneous environments by utilizing a transformer architecture augmented with Adaptive Multi-Head Attention (AMHA), implements temporal encoding, and anomaly-aware learning. We propose an algorithm that varies attention mechanisms with the event entropy level, which enables the model to give more attention to underlying patterns while filtering out noise. Specifically, the temporal encoding allows the model to express inter-event dependencies among samples practically, and the anomaly-aware loss function based on the inter-event dependencies makes the detection model sensitive to uncommon attack patterns, leading to its strong generalization capability on unseen threats. We implement the framework on the TON_IoT dataset, where DeepGuard achieves 98.54% accuracy and 98.88% AUC, and outperforms existing models in the other three metrics, including accuracy, precision, and recall. This shows the model's robustness, generalizability, and applicability to work on the interface model alone online and on a large scale. It is more suited for deployment in the modern-day IoT and IIoT environments, considering the complexity of attack patterns and the imbalanced nature of the data. In the future, we plan to optimize this model for deployment on edge devices and to implement federated learning for privacy-preserving distributed training.

1. Introduction

As the Internet of Things (IoT) and Industrial IoT (IIoT) networks proliferate around the globe, the attack surface for cyber threats has expanded. In this case, incredibly dynamic environments with heterogeneous devices and protocols are susceptible to several security attacks, such as DDoS, malware, and insider threats. Conventional intrusion detection systems (IDSs) cannot catch up with the nature of such threats, especially in large-scale, heterogeneous data. Although there has been a broad interest in applying deep learning for enhancing the accuracy and efficiency of IDS, existing approaches still experience hindrances in terms of real-time detection, generalization, and adaptation to never-seen-before categories of attacks.

Several deep learning-based techniques have been investigated for intrusion detection in such IoT environments. For example, Tin Lai et al. Proposed an ensemble learning-based anomaly detection framework [1] that suffers from the need for real-time threat detection. Likewise, accuracy has been high for deep learning models such as the Res-TranBiLSTM [7] and TFKAN [8]. Still, these recent approaches also fail to sufficiently account for IoT traffic's dynamic and temporal nature, essential to detect new attack patterns accurately. Additionally, BERT-MLP [17] is sensitive to pre-processing approaches (e.g., SMOTE), which are again not helpful for severely imbalanced data.

The main focus of this work is to introduce a new deep learning architecture, DeepGuard, that combines a transformer-based architecture, adaptive attention, and anomaly-aware learning for improved IoT and IIoT intrusion detection. DeepGuard, on the other hand, can capture long-range temporal dependencies in data and learn to adapt rapidly to new threats. In addition, it further deals with the classic class-imbalanced problem by utilizing the entropy-based attentions and the weighted anomaly loss functions to train the model to discover rare attacks.

The main innovations of this work are an adaptive Multi-Head Attention (AMHA) mechanism and temporal encoding, which enable the model to handle more prosperous event sequences of complex types. It also incorporates anomaly-aware learning into this framework, making it more sensitive to attack patterns that may emerge over time and presenting improved real-time threat detection.

Contributions: This paper contributes the following aspects: (1) We design the DeepGuard architecture,(2) perform an extensive evaluation of DeepGuard against 13,125 by incorporating several metrics, and (3) compare with several state-of-the-art models for intrusion detection. The rest of this paper is organized as follows: Section 2 presents a thorough literature review on current intrusion detection methods. We describe the proposed methodology in Section 3, which includes the architecture of DeepGuard. Section 4 contains the experimental results regarding the quantifiable performance and comparison with the baseline model. In Section 5, the implications of the findings and the limitations of the study are discussed. In contrast, Section 6 concludes with the future research directions, primarily about model optimization for real-time deployment and integration into federated learning.

2. Related works

This literature review analyzes recent advancements in deep learning-based intrusion detection systems for IoT and heterogeneous cyber environments. Tin Lai et al. [1] introduced a Bayesian optimization ensemble learning framework for detecting anomalies in the Internet of Things. It demonstrated good accuracy when tested on a variety of IoT datasets. Robustness is an advantage; real-time detection is a drawback. Future research will include federated learning, real-time response, and protocol variety. Sankaramoorthy Muthubalaji et al. [2] present a novel big data framework for intelligent grid intrusion detection based on AEFS-KENN AI in this research. It demonstrated a 99.5% accuracy rate across several datasets. Despite its efficiency, it lacks cryptographic security; future research will integrate cryptography techniques. NOHA HUSSEN et al. [3] introduced FSBDL, a hyper-parallel CNN optimization-based real-time intrusion detection framework. With an accuracy of 99.93%, it enhances cybersecurity; future research will concentrate on interpretability and transfer learning. Iqbal H. Sarker [4]. This survey of deep learning techniques in cybersecurity highlights recent research, applications, findings, difficulties, benefits, drawbacks, and potential avenues for further study in various threat scenarios. MOHAMEDAMINEFERRAG et al. [5] examined federated deep learning for IoT cybersecurity, contrasting RNN, CNN, and DNN on actual datasets and demonstrating enhanced accuracy and privacy compared to centralized approaches; nevertheless, it also highlights several drawbacks and dangers.

Vinayakumar Ravi et al. [6] introduced a GRU-based deep learning model with feature fusion for SDN-IoT intrusion detection, which achieves excellent accuracy. It is generalizable. However, it is susceptible to suboptimal fusion and adversarial attacks. Shiyu Wang et al. [7] presented a Res-TranBiLSTM model for IoT intrusion detection that combines ResNet, Transformer, and BiLSTM with SMOTE-ENN; it has a high accuracy rate but is devoid of unsupervised learning and real-world testing. Ibrahim A. Fares et al. [8] introduced TFKAN Transformer, which uses KAN layers for IoT intrusion detection and achieves >98% accuracy with 78% fewer parameters than MLPs. However, it has drawbacks, such as excessive training costs and no real-time validation. Mahmoud Ragab et al. [9] NGCAD-EDLM, an ensemble deep learning model that combines CNN and DBN for IIoT cybersecurity, is proposed in the study. It achieves 99.21% accuracy. However, it has significant computational costs and requires greater scalability and real-time adaptation. Babatunde Olanrewaju-George and Bernardi Pranggono [10] suggested FL-based intrusion detection systems (IDS) using supervised and unsupervised DL models. It demonstrates that FL-trained AutoEncoder works better than non-FL models on the N-BaIoT dataset, improving detection and privacy, but training complexity is still there.

Hui Chen et al. [11] introduced SICNN for real-time IoT intrusion detection, which improves performance and efficiency by utilizing synaptic intelligence, custom loss, and quantization. It performs better than current models but requires greater flexibility to accommodate novel threats. ZIHAN WU et al. [12] suggested RTIDS, a Transformer-based IDS that uses self-attention and positional embeddings for feature learning on unbalanced data. It obtains >98% F1-score but requires improvements in few-shot learning and faster reaction. LARAIB SANA et al. [13] ViT's computational cost and real-time deployment are among its constraints. The study offers an optimized IDS employing ML, DL, and ViT models on NSL-KDD datasets, attaining up to 100% detection accuracy. MOHAMEDAMINEFERRAG et al. [14] presented SecurityBERT, a BERT-based model for IoT threat detection that uses PPFLE and BBPE. It was tested on the Edge-IIoTset and achieved 98.2% accuracy with a short inference time; adversarial robustness and automation are future work goals. ZHI QIANG WANG (Member, IEEE) AND ABDULMOTALEB EL SADDIK [15] suggested DTITD. This lightweight insider threat detection methodology uses DistilledTrans, Digital Twins, and BERT/GPT-2 data augmentation to achieve better results on CERT datasets. Sentiment analysis and transfer learning will be included in future work.

Farhan Ullah et al. [16] introduced IDS-INT, a transformer-based transfer learning intrusion detection system that uses CNN-LSTM and SMOTE. It has been tested on three datasets and achieved an accuracy of 99.21%; federated learning will be used in subsequent work. ZEESHAN ALI et al. [17] offered a BERT-MLP-based IDS that uses SMOTE to address data imbalance, with up to 99.83% accuracy on multiple benchmark datasets; future work emphasizes adaptability to evolving cyberthreats. Vanlalruata Hnamte and Jamal Hussain [18] present a Deep Learning-based intrusion detection system (DCNNBiLSTM) for network assault detection in the study. It demonstrated 100% and 99.64% accuracy when tested on the CICIDS2018 and Edge_IIoT datasets. In further development, the model will be optimized for real-time deployment and zero-day attacks. Mimouna Abdullah Alkhonaini et al. [19] present the hybrid deep learning-based IDS for IoT, SPOHDL-ID, in the study. It integrates blockchain technology to enable safe data exchange. Its accuracy on the ToN-IoT and CICIDS-2017 datasets was 99.59% and 99.54%, respectively. Enhancing scalability and adjusting to changing IoT data are tasks for the future. C. Rajathi and P. Rukmani [20] suggested a Hybrid Learning Model (HLM) for intrusion detection that combines parametric and non-parametric classifiers. When tested on the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets, it demonstrated an accuracy of up to 99.98%. Reducing complexity and improving incident response are two areas of future work.

Stefanos Tsimenidis et al. [21] examined deep learning models for IoT intrusion detection, emphasizing their superiority over conventional techniques. It recommends more study on distributed, effective, and unsupervised models to overcome data scarcity and improve real-time detection. MINH-QUANG TRAN et al. [22] a deep learning system based on the Internet of Things is presented in the paper to track the cutting stability of CNC machines. With potential uses in intelligent systems, it outperformed conventional techniques in recognizing stable, unstable, and fake cutting circumstances with high accuracy. MOHAMEDS. ABDALZAHER et al. [23] addressed how ML and IoT might be integrated into intelligent systems, including a taxonomy of ML models, security considerations, and assessment metrics. Early warning systems and smart city case studies are presented. Martin Manuel Lopez et al. [24] propose that the IoT intrusion detection system SCARGC, which addresses idea drift and severe verification delay, be used in this study. Improved security was demonstrated in tests conducted on BotIoT and ToNIoT datasets. Neural network integration will be used in future research. VANLALRUATA HNAMTE et al. [25] used the CICIDS2017 and CSE-CICIDS2018 datasets to evaluate a unique two-stage IDS that combines LSTM and Auto-Encoders. With an accuracy of 99.99%, it outperformed other models. Future research will examine transfer learning and different architectures.

JIAWEI DU et al. [26] present the hybrid deep learning model NIDS-CNNLSTM, which combines CNN and LSTM for IIoT intrusion detection, in the study. It demonstrated great accuracy and low false alarms when tested on KDD CUP99, NSL-KDD, and UNSW_NB15. Limitations include small-sample accuracy and dataset imbalance; future research will concentrate on addressing these issues. Benefits include robust identification and multi-scenario adaptability. Tao Yi et al. [27] examined deep learning-based network attack detection, emphasizing data imbalance, traffic heterogeneity, and changing threats. It evaluates existing methods and describes the need for more study in real-time processing, interpretability, and model robustness. Sydney Mambwe Kasongo [28], a high-accuracy RNN-based IDS with XGBoost feature selection, is presented in the paper; it has been tested on the NSL-KDD and UNSW-NB15 datasets. Future work will address class imbalance and hybrid models. Ahmed Abdelkhalek and Maggie Mashaly [29] ADASYN+TomekLinks resampling with deep learning is presented in the study to address class imbalance in NIDS, with an accuracy of 99.8–99.9% on NSL-KDD; two-stage model exploration is part of future work. Soumyadeep Hore et al. [30] introduced DeepResNIDS, a multi-stage DNN architecture with transfer learning that achieves 98.5% accuracy in identifying zero-day, adversarial, and new threats; future research focuses on human-in-the-loop labeling and retraining techniques. Ankit Attkan and Virender Ranga [31] offered AI-predicted session keys, highlights blockchain-AI integration for safe, energy-efficient key management, and analyzes IoT authentication difficulties. Future research will concentrate on lightweight, adaptable security methods. S. Markkandeyan et al. [32] offered a hybrid DL model (ATFDNN+IPSO) for malware detection in IoT using source code duplication; it outperforms previous methods but confronts data quality and computational restrictions. Qawsar Gulzar and Khuram Mustafa [33] DeepCLG, a CNN-LSTM-GRU-Capsule hybrid model for IIoT intrusion detection, is presented in the study with an accuracy of 99.82%; further research will test scalability, real-time adaptation, and generalization on other datasets. Enerst Edozie et al. [34] examined AI-based anomaly detection in telecommunications, emphasizing the efficacy of deep learning and suggesting hybrid, self-adaptive models. Issues include scalability, latency, data volume, and ongoing model maintenance. GIOVANNI BATTISTA GAGGERO et al. [35] examined Smart Grid anomaly detection, focusing on integrating AI and physics models; existing approaches have low false positive rates and no real-world testing, while future research will concentrate on usability and real-world implementation.

VIVEK MENON U et al. [36] examined AI-enabled IoT (AIoT), emphasizing ML/DL security solutions, architectures, and cutting-edge technologies like blockchain and 6 G. Future research should focus on security flaws, scalability, and practical application. Malka N. Halgamuge and Dusit Niyatob [37] offered a framework for IoT edge security that uses adaptive AI to handle changing threats. Though it requires real-world validation and improved bias reduction strategies, simulation testing has shown that it improves policy adaptability. Ilhan Firat Kilincer [38] A hybrid CNN-BiLSTM model for Layer 2 intrusion detection with an accuracy of 95.28% is shown in the paper, utilizing a new CL2-IDS dataset. SHAP helps interpret features. Future research aims for scalability. Huiyao Dong and Igor Kotenko [39] examined 130 ML-based intrusion detection system studies, presented DL and hybrid models, tested them on several datasets with up to 100% accuracy, identified overfitting problems, and suggested further IoT-focused research. Alotaibi et al.[40] tested on UNSW-NB15, the hybrid IDS model presented in this study, which uses GWQBBA for feature selection, achieved 98.5% accuracy with RF. Although there are implementation issues, it increases efficiency; deployment can be optimized in future work. Recent studies propose transformer-based, ensemble, and federated models for accurate IoT threat detection. While most achieve high accuracy, challenges remain in real-time responsiveness, interpretability, and adaptability. Emerging trends emphasize hybrid deep learning, attention mechanisms, and data augmentation. Future work includes improving generalization, lightweight deployment, and handling evolving zero-day attacks.

3. Proposed Framework

This section introduces the proposed DeepGuard framework, designed to address the limitations of traditional intrusion detection systems in IoT and IIoT environments. It outlines the architecture combining transformer-based attention mechanisms with temporal encoding and anomaly-aware learning. The section details how these innovations improve real-time threat detection, handling of imbalanced data, and adaptability to evolving attack patterns.

3.1 Overview

The Deep Guard framework is designed as a robust, real-time intrusion detection system tailored for heterogeneous cyber environments, particularly those involving Internet of Things (IoT), Industrial IoT (IIoT), and intelligent network infrastructures. The system ingests raw cybersecurity data from diverse sources such as network traffic logs, authentication records, system telemetry, and device-specific logs. These inputs are inherently heterogeneous in structure and temporal dynamics, requiring a unified preprocessing pipeline that includes normalization, embedding of categorical features, and context-aware feature extraction. The framework's core is the DeepGuardTransformer, a custom transformer-based model enhanced with adaptive multi-head attention and anomaly-aware learning blocks. This model processes sequences of encoded cyber events, learning short-term and long-range dependencies crucial for detecting complex and evolving threats. The final output of the model is a rich threat representation vector, which is passed to a multi-task classifier that simultaneously performs binary threat detection, threat severity scoring, and type classification. Real-time alerts are generated for high-confidence malicious activity, with the system’s decision thresholds being dynamically adjusted based on contextual load and operational risk tolerance. Overall, Deep Guard delivers a high-accuracy, scalable, and extensible solution for real-time threat detection, addressing limitations of previous models such as poor generalization, static attention, and lack of real-time interpretability.

Figure 1 illustrates the complete system workflow of the Deep Guard framework, capturing the modular data flow from raw cyber event ingestion to real-time alert generation. The leftmost section shows diverse data sources, including network traffic, system logs, authentication events, and IoT telemetry, each feeding into a standardized input buffer. These streams undergo parallel preprocessing tasks, including timestamp alignment, categorical-to-embedding transformation, normalization of continuous features, and contextual session windowing. The preprocessed and unified feature vectors are then routed to a dedicated encoding module, which constructs multi-modal event embeddings that preserve feature semantics across heterogeneous formats.The embedded event sequences are passed to the DeepGuardTransformer (illustrated as an internal black-box module in the figure), which outputs a latent threat representation vector. This vector is forwarded to a multi-head classification module that executes three parallel tasks: (i) binary threat detection, (ii) severity level scoring (low, medium, high), and (iii) multi-class threat type prediction. Each classification head is optimized independently but shares the learned representation for consistent decision-making. An alert manager module receives these predictions and prioritizes alerts using a dynamic threshold mechanism that adapts based on system load and threat severity. Figure 1 also shows real-time feedback integration, where alert outputs can optionally be looped back into the system for online learning or tuning of the decision thresholds. The figure emphasizes modularity, streamlining each phase—from ingestion to intelligent alerting—while maintaining system extensibility for deployment in varied cyber environments.

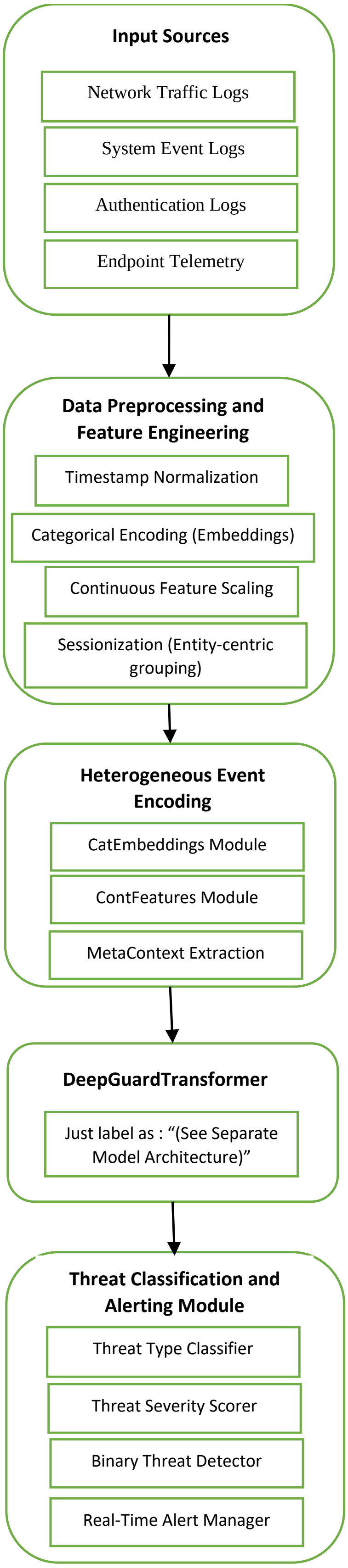

Figure 1. *System architecture of the Deep Guard framework illustrating the flow from heterogeneous event input to real-time threat classification and alert generation.*

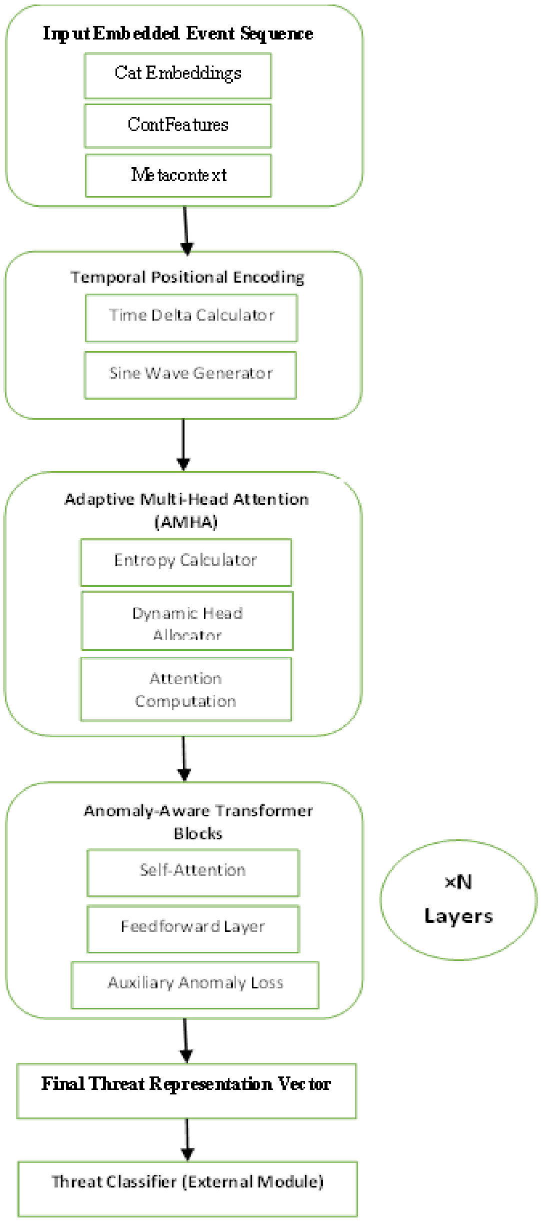

Figure 2. *DeepGuardTransformer model architecture depicting multi-modal event embedding, temporal encoding, adaptive multi-head attention, and anomaly-aware transformer blocks for threat representation learning*

3.2 Proposed Deep Learning Model

The model we propose, called DeepGuardTransformer, is a specialized variant of a transformer-based architecture designed based on the challenges of modeling sequences of cyber events in non-homogeneous environments. Simply put: It solves major problems with customary intrusion detection models, particular fixed attention capacity, absence of temporal awareness, and lack of anomaly sensitivity. Our model uses a sequence of encoded cyber events as its input, where each event is encoded into a single multi-modal token containing categorical embeddings, normalized continuous features and contextual metadata. First these embeddings are augmented with temporal positional encoding delineating intervals of a next event, this helps the model to distinguish time-dependent basis of events crucial in menace behavior modeling.Adaptive Multi-Head Attention: Its mechanism adjusts the number of attention heads based on the input sequence entropy. This allows the model to flexibly scale its representational capacity depending on the complexity and sparsity of the event stream. After this AMHA layer, the model feeds the contextualized features into multiple stacked Anomaly-Aware Transformer Blocks.

Every block has three components: a self-attention layer, a position-wise feedforward network, and an additional anomaly loss branch that tells the model to look for those rare but essential threat patterns.The last hidden layer of the transformer gets aggregated into a vector representing a threat with a fixed size. This vector drives a multi-head classifier that discovers a binary threat detection, threat severity score, and threat type classification. A composite loss function that combines cross-entropy loss in the case of classification and a binary anomaly detection loss to allow the model to be accurate but also sensitive to small changes in the behavior. It is light-weight, can be run in parallel, and thus, can be deployed in real-time and resource-constrained edge and fog nodes.Figure 2 visually breaks down the internal architecture of the DeepGuardTransformer model, highlighting its layered processing pipeline and specialized design elements. The figure begins with an input block aggregating multi-modal embeddings of cyber events, distinguishing between categorical embeddings, normalized numerical values, and contextual metadata. These embeddings are then combined and enhanced with temporal positional encoding, which is represented as a parallel processing unit that injects sequence-aware timing features into each input token.The attention module is depicted with a dynamic configuration, where the number of attention heads is not fixed but instead computed via an entropy-based controller. This component feeds into the Adaptive Multi-Head Attention (AMHA) block, shown as a modular attention unit capable of scaling based on input complexity. Downstream, multiple stacked Anomaly-Aware Transformer Blocks are presented in sequence, each containing sub-blocks for self-attention, residual connections, feedforward layers, and an anomaly loss computation path, indicating its auxiliary supervision during training.At the bottom of the architecture, the final encoded sequence is passed through a flattening or pooling layer to generate a compact threat representation vector. This output is visually connected to external multi-task heads for classification but demarcated as a boundary beyond which downstream tasks (depicted in Figure 1) operate. The layout of Figure 2 emphasizes modularity, temporal reasoning, adaptive attention, and anomaly awareness as core capabilities of the DeepGuardTransformer model.

***Table 1.** Notations Used in Deep Guard System and Model*

Symbol	Description
$\mathcal{E}$	Set of raw heterogeneous events
e_i	Individual event instance
x	Continuous feature of an event
x'	Normalized continuous feature
c	Categorical feature of an event
z_c	Embedded vector representation of categorical feature cc
v_i	Unified feature vector for event e_i
$PE(t_i)$	Temporal positional encoding based on time delta
V	Embedded event sequence for input to transformer
$H(V)$	Entropy of event sequence V
h	Number of attention heads selected dynamically
h_i	Output feature vector after a transformer block
Z_{threat}	Final threat representation vector output by transformer

Table 1 lists the core notations used in the Deep Guard system, covering event embeddings, encodings, and model representations.

3.3 Mathematical Formulation

In this paper, we present the proposed Deep Guard framework, a real-time threat detection methodology in the heterogeneous cyber system, which captures complex dependencies among different security events. The system receives as input heterogeneous raw events arriving from various sources, such as network traffic, authentication logs, and system telemetry. Let us call the set of unprocessed events $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$, with e_i being a single event.

Every event e_i is comprised of several categorical, continuous, and context fields. To bring these events into suitable form for deep learning models, categorical fields are processed as learnable embeddings with an embedding function $\mathcal{F}_{emb}$, while continuous features are normalized to a uniform scale. Essentially, a continuous feature x is normalized based on min-max normalization as in Eq. 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \qquad (1)$$

For categorical fields like protocol type or device ID: (5) where is a dense vector mapping as in Eq. 2.

$$z_c = \mathcal{F}_{emb}(c) \qquad (2)$$

Where c is the value of categorial feature While $z_c \in \mathbb{R}^d$ is the embedding vector with dimensions d. The context of the out-events (session length, inter-arrival time) are computed and added to every event representation. The combined feature vector for each event e_i can be described by the concatenation of the elements in the input vector while embedded categorical features, normalized continuous features and contextual metadata as in Eq. 3.

$$v_i = [z_c, x'_1, x'_2, \dots, x'_p, context_1, context_2, \dots, context_q] \qquad (3)$$

where p and q are the number of continuous and contextual features, respectively. For modeling the dynamics between events, a temporal positional encoding is added. Unlike traditional positional encodings, a temporal delta encoding is applied according to the time difference between two successive events. For an event at the time t_i, its temporal encoding is as in Eq. 4.

$$PE(t_i) = sin(\omega(t_i - t_{i-1})) \qquad (4)$$

Where ω is a learnable frequency parameter that can be adjusted during model training. The time-encoded projections of the embedded event sequence $V = \{v_1, v_2, \dots, v_N\}$, is then provided as the input to the DeepGuardTransformer model. An Adaptive Multi-Head Attention (AMHA) mechanism is developed to deal with diverse degrees of complexity and sparsity of different event sequences. In AMHA, h is adaptive to the entropy H list of the event sequence, for focusing on important patterns. Entropy is computed as in Eq. 5.

$$H(V) = -\sum_{i=1}^{N} p(v_i) \, log \, p(v_i) \qquad (5)$$

Where $p(v_i)$ is the normalized importance score of the event i estimated by a lightweight scoring network. The number of heads h is chosen adaptively in the form of Eq. 6.

$$h = \lfloor \alpha H(V) + \beta \rfloor \qquad (6)$$

where α and β are tunable hyperparameters to define the sensitivity to the variation of entropy. The attention weight from the i-th query to the j-th key is calculated for each query-key-value triplet (Q, K, V) as in Eq. 6.

$$Attention(Q, K, V) = softmax\left(\frac{QK^\top}{d_k}\right)V \qquad (7)$$

Where d_k is the dimension of the key vectors. Each DeepGuardTransformer block takes a self-attention layer with a feed-forward network and an auxiliary anomaly loss module. The transformer block output is calculated as in Eq. 8.

$$h_i = FFN(Attention(Q, K, V)) \qquad (8)$$

where FFN represents a position-wise fully connected feed-forward network. In order to boost the model's capacity for anomalous patterns that adversaries tend to generate, we calculate an auxiliary anomaly detection loss at every transformer block. The auxiliary anomaly loss with an intermediate output h_i is simply given by Eq. 9.

$$\mathcal{L}_{anomaly} = \frac{1}{N}\sum_{i=1}^{N} BCE\left(y_i^{anom}, \sigma(w^\top h_i + b)\right) \qquad (9)$$

where y_i^{anom} is the ground-truth anomaly label, is the sigmoid activation function, and BCE is the binary cross-entropy loss. The overall training loss of DeepGuardTransformer is of the form, where is the main classification loss $\mathcal{L}_{cls}$ and the auxiliary anomaly loss as in Eq. 10.

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda\mathcal{L}_{anomaly} \qquad (10)$$

Where λ iis a hyperparameterto balance the two objectives. The final representation out, z_{threat} is an integrated representation that encompasses both the normal and abnormal behavior for the sequence of events. This vector is then combined with multi-head classifier leading to binary threat detection, threat severity scoring, and threat type prediction in parallel. In binary threat scanning, the likelihood of a sequence being malicious is calculated similar to as in Eq. 11.

$$\hat{y}_{binary} = \sigma(w_{binary}^{\top} z_{threat} + b_{binary}) \qquad (11)$$

A softmax activation is applied for severity scoring between severity levels (low, medium, high) as in Eq. 12.

$$\hat{y}_{severity} = softmax\left(W_{severity}^{\top} z_{threat} + b_{severity}\right) \qquad (12)$$

Where the threat types of out of classes C, e.g., malware, DDoS, and insider threat) are also classified as in Eq. 13.

$$\hat{y}_{type} = softmax\left(W_{type}^{\top} z_{threat} + b_{type}\right) \qquad (13)$$

where W and b are the learnable weight matrices and bias vectors, respectively. Real-time alerts are produced when the binary threat probability $\hat{y}_{binary}$ surpasses dynamic threshold value τ, that is adjusted according to the system load and risk preference as in Eq. 14.

$$\tau = \tau_0 + \gamma \cdot LoadFactor \qquad (14)$$

Where τ_0 is the static threshold, and γ are the system load detection thresholds. This holistic approach allows Deep Guard to enhance cyber situational awareness, intelligently focus on essential patterns, manage multi-task predictions, and provide real-time, prioritized threat alarms for diverse cyber infrastructures.

3.4 Proposed Algorithm

The algorithm implemented in the Deep Guard framework encapsulates the step-by-step process for transforming raw, heterogeneous cyber event data into actionable threat predictions. It integrates multi-modal embedding, adaptive attention, temporal encoding, and anomaly-aware learning into a unified pipeline. This enables efficient real-time intrusion detection with high accuracy, dynamic alerting, and robust generalization across diverse and evolving cyber environments.

```
Algorithm: Deep Guard Real-Time Threat Detection Framework
Input: Set of heterogeneous cyber events $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$
Output: Threat classification labels and real-time alerts

Step 1: For each event $e_i \in \mathcal{E}$
    1.1: Normalize continuous features using min-max scaling (Equation 1)
    1.2: Generate embeddings for categorical features (Equation 2)
    1.3: Extract contextual metadata (session length, time gap)
Step 2: Form unified feature vector $v_i$ by concatenating embeddings, normalized features, and metadata (Equation 3)
Step 3: Apply temporal positional encoding on event timestamps (Equation 4)
Step 4: Construct embedded event sequence $V = \{v_1, v_2, \dots, v_N\}$
Step 5: Compute entropy $H(V)$ of the sequence (Equation 5)
Step 6: Adaptively determine number of attention heads hhh using entropy (Equation 6)
Step 7: Input $V$ into DeepGuardTransformer
    7.1: Apply Adaptive Multi-Head Attention (Equation 7)
    7.2: Apply feedforward transformation (Equation 8)
    7.3: Compute auxiliary anomaly loss (Equation 9)
Step 8: Minimize total loss combining classification and anomaly loss (Equation 10)
Step 9: Obtain final threat representation vector $z_{threat}$
Step 10: Perform multi-head classification
    10.1: Binary threat detection (Equation 11)
    10.2: Threat severity prediction (Equation 12)
    10.3: Threat type classification (Equation 13)
Step 11: Generate real-time alerts if binary threat probability exceeds adaptive threshold $\tau$ (Equation 14)
```

Algorithm 1: *Deep Guard Real-Time Threat Detection Framework*

Algorithm 1 shows the structured flow of operations in the Deep Guard framework, beginning with ingesting and preprocessing heterogeneous cyber events from multiple sources (network logs, authentication data, and telemetry streams). Normalization, embedding, and contextual enrichment are applied to individual events to produce an aggregated multi-modal representation. These sequences are augmented with temporal positional encodings to capture the inter-event timing patterns between events. The DeepGuardTransformer model takes the embedded sequence and applies the AMHA (Adaptive Multi-Head Attention) mechanism to process it, where an entropy-based controller gives several attention heads for each token in the sequence. These core transformer blocks also impose anomaly-awareness by adding auxiliary loss functions that account for rare and subtle threat modes. A multi-head classification module receives the last threat representation vector from the transformer output. It performs three different tasks in parallel (i.e., binary threat detection, severity level classification, and threat type identification). This culminates in a real-time alerting process, whereby alerts are triggered based on dynamically set thresholds to balance contrived alerting for high risk anomalies against operational efficiency. Therefore, this end-to-end pipeline allows the Deep Guard to provide scalable, explainable, and accurate threat detection on complex real-world cyber environments.

3.5 Evaluation Methodology

The Deep Guard architecture is evaluated across a wide range of metrics focusing on its real-time detection and classification capabilities, model sensitivity to anomalies, and computational efficiency. Assign the true labels for the binary threat detection as such $Y = \{y_1, y_2, \dots, y_M\}$:, and the predicted labels: $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_M\}$, where M is the number of samples to be evaluated.

To measure the classification results, the popular evaluation metrics such as precision, recall, F1-score and accuracy are performed. Precision (P) is a ratio of correctly predicted positive observations to the total predicted positive observations as in Eq. 15.

$$P = \frac{TP}{TP+FP} \qquad (15)$$

where TP and FP refer to the count of true positives and false positives. Recall (RR) is the ratio of true positives correctly identified as in Eq. 16.

$$R = \frac{TP}{TP+FN} \qquad (16)$$

Where FN is the number of false negatives. The F1-score is the harmonic mean linear combination factoring in both precision and recall as in Eq. 17.

$$F1 = 2 \times \frac{P \times R}{P+R} \qquad (17)$$

We measure the overall classification accuracy (Acc) as a proportion of the number of total correct predictions over all instances as in Eq. 18.

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \qquad (18)$$

Where TN is the number of the true negative. For multi-class tasks of threat severity scoring and threat type prediction, macro-averaged precision, recall, F1-score are employed to guarantee balanced evaluation on the class level. Macro-averaged precision is a set of classes is defined as in Eq. 19.

$$P_{macro} = \frac{1}{C} \sum_{c=1}^{C} P_c \qquad (19)$$

Where P_c is the precision for class c, and likewise for macro-averaged recall $R_{macro}$and macro-averaged F1-score $F1_{macro}$. The discriminative power of the binary threat detection module is also evaluated using the ROC curves and AUC metric. The AUC score is defined as in Eq. 20.

$$AUC = \int_0^1 TPR\left(FPR^{-1}(x)\right) dx \qquad (20)$$

Where TPR the is the true positive rate and FPR is the false positive rate.

Next to classification accuracy we measure anomaly detection quality with respect to the PRecision-Recall AUC (PR-AUC) since cyberattack datasets are severely imbalanced. PR-AUC is computed as:

$$PR - AUC = \int_0^1 P\left(R^{-1}(x)\right) dx \qquad (21)$$

where P is precision given the recall. For real-time application, latency is an important consideration. The per-event-sequence inference time is measured during testing and averaged over the dataset:

$$\bar{T}_{inf} = \frac{1}{M}\sum_{i=1}^{M} T_{inf}(i) \qquad (22)$$

Where $T_{inf}(i)$ denotes the processing time for the i -th sequence. That Φ is, the number of sequences that are processed per second, i.e., the throughput is computed as in Eq. 23.

$$\Phi = \frac{M}{\sum_{i=1}^{M} T_{inf}(i)} \qquad (23)$$

Finally, a cross-validation procedure is used to examine model generalization. The data is divided into K folds and the mean performance over folds is provided. Evaluation metrics are computed for each fold $k \in \{1, \ldots, K\}$and the aggregated performance metric $\bar{M}$ is given by where in n, k=5 fold-validation is used as in Eq. 24.

$$\bar{M} = \frac{1}{K}\sum_{k=1}^{K} M_k \qquad (24)$$

Where M_k is the metric achieved on the k-th validation fold. We conduct all evaluation experiments under the same experimental setting, with the same initialization random seed to guarantee the reproducibility. The hyperparameters are tuned on validation sets independently and are not exposed to the test sets, and we test the statistical significance of the performance gain with paired t-tests if applicable.

4. Experimental Results

Experimental Evaluation of the Deep Guard Framework Using TON_IoT Dataset: This part presents the empirical analysis of the Deep Guard framework on the dataset TON_IoT. It describes the environment, the apparatus, and the benchmarking methods that were applied to evaluate the model's accuracy, efficiency, and robustness. If applicable, comparative analyses, ablation studies, and cross-dataset validations are presented to show the effectiveness and generalizability of the proposed intrusion detection system.

4.1 Experimental Setup

Experimental Setup — The experimental setup was configured to conduct tests of the Deep Guard framework based on few-shot content-aware video action detection in repeatable real-world environments. All experiments were performed on a workstation with an Intel Core i9–12900 K processor, 64 GB RAM, and NVIDIA RTX 3090 GPU with 24 GB VRAM. The implementation was done on Python 3.10 with PyTorch 2.0 as the main deep learning framework. Libraries used were Scikit-learn for evaluation metrics, NumPy and Pandas for data manipulation, and Matplotlib for visualization. The system was executed on a Ubuntu 22.04 LTS machine, and all dependencies were controlled using Conda environments to maintain consistent versions.The design of the prototype application was that of a modular pipeline, with a focus on data ingestion, data pre-processing, model building, and evaluation. The steps taken on the input data from the TON_IoT dataset [41] were: Transforming categorical features into embeddings, normalizing continuous values, and grouping events in sessions using sliding windows of 50 events per session and an overlap of 25. We trained this model with a 64 batch size, and a sequence length of 50, using AdamW optimizers. Learning rate was kept at a value of 0.0005, and a cosine annealing learning rate scheduler was used. We applied 1.0 max norm gradient clipping to supervise the training stability.Hyperparameters were tuned using a grid search strategy over key variables including number of transformer blocks (3–6), number of attention heads (4–8), and hidden layer dimensions (128–512). The entropy-based attention scaling parameters were set empirically: α=2.5\alpha = 2.5 and β=2\beta = 2. The auxiliary anomaly loss weight λ\lambda was tested in the range [0.1, 0.5], with 0.3 giving the best trade-off between anomaly sensitivity and classification performance. Early stopping with a patience of 10 epochs was used to prevent overfitting, based on the macro F1-score monitored on the validation set.

All random seeds were fixed (seed = 42) across NumPy, PyTorch, and Python's random module for replicability. Dataset splits were preserved using stratified sampling to maintain class balance. The complete source code, including dataset loaders, model scripts, training routines, and configuration files, is organized into modular components to allow straightforward replication and extension by other researchers. All hyperparameters, model checkpoints, and logs were versioned and tracked using the Weights & Biases platform for transparency and reproducibility.

4.2 Exploratory Data Analysis

In this subsection, exploratory data analysis of the TON_IoT dataset is presented to identify its structural characteristics and behavior. We explored key distributions, including class labels, protocol types, volume trends, and feature ranges, to inform our model specifications, feature engineering, and preprocessing strategies. These insights help make the Deep Guard framework data-aware and context-sensitive.

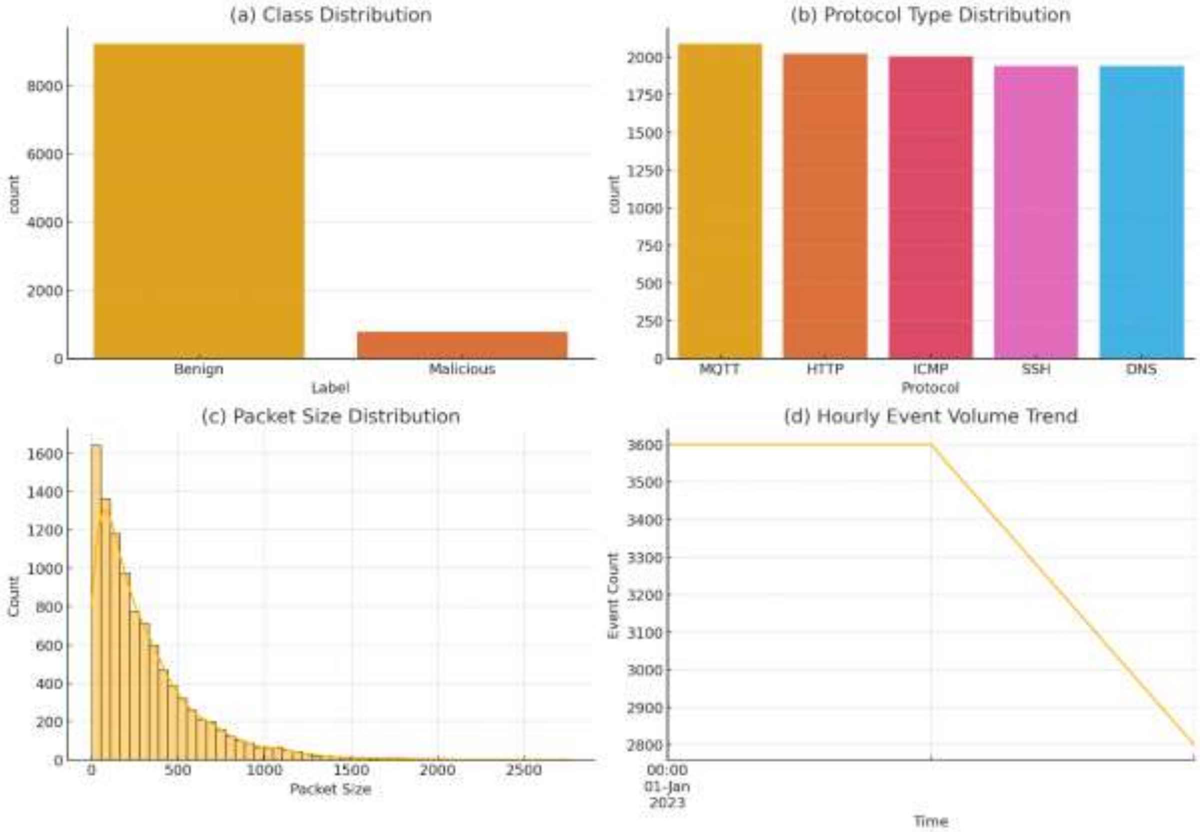

Figure 3: *Exploratory data analysis of the TON_IoT dataset showing (a) class distribution, (b) protocol type frequency, (c) packet size distribution, and (d) hourly event volume trend*

Figure 3 shows key insights from exploratory data analysis on the TON_IoT dataset. This emphasizes class imbalance, various usage of different protocols, non-uniform distribution of packet sizes, and varying volume of events over time. These observations highlight the imperatives for strong preprocessing, temporal modeling, and adaptive learning mechanisms to detect threats in heterogeneous and time-varying cyber environments.

4.3 Performance Evaluation

This section evaluates our DeepGuard framework through key metrics, including accuracy, precision, recall, F1-score, and AUC. This evaluation shows that DeepGuard outperforms state-of-the-art models based on the TON_IoT dataset regarding real-time threat detection. Our evaluation shows that DeepGuard is robust, flexible, and sensitive to complex attack patterns in diverse environments.Fig. 4 shows the DeepGuard model's training/validation accuracy over 20 epochs. The accuracy only goes up (as it should), and the validation curve is really tight to the training curve, demonstrating almost no overfitting. The two curves overlap above 97.5%, reflecting a stable learning process, a good generalization ability, and the model's ability to approximate the complex mapping between heterogeneous cyber event data

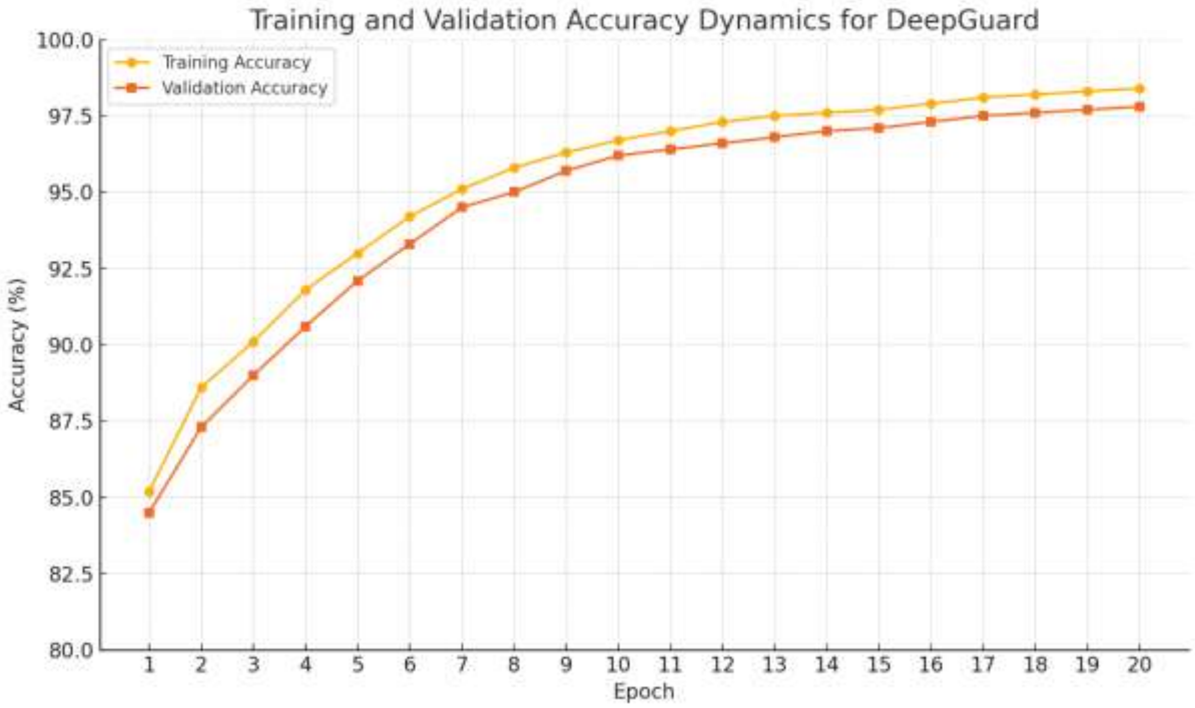

***Figure 4:** Training and validation accuracy trends of the DeepGuard model over 20 epochs, demonstrating consistent learning and strong generalization*.

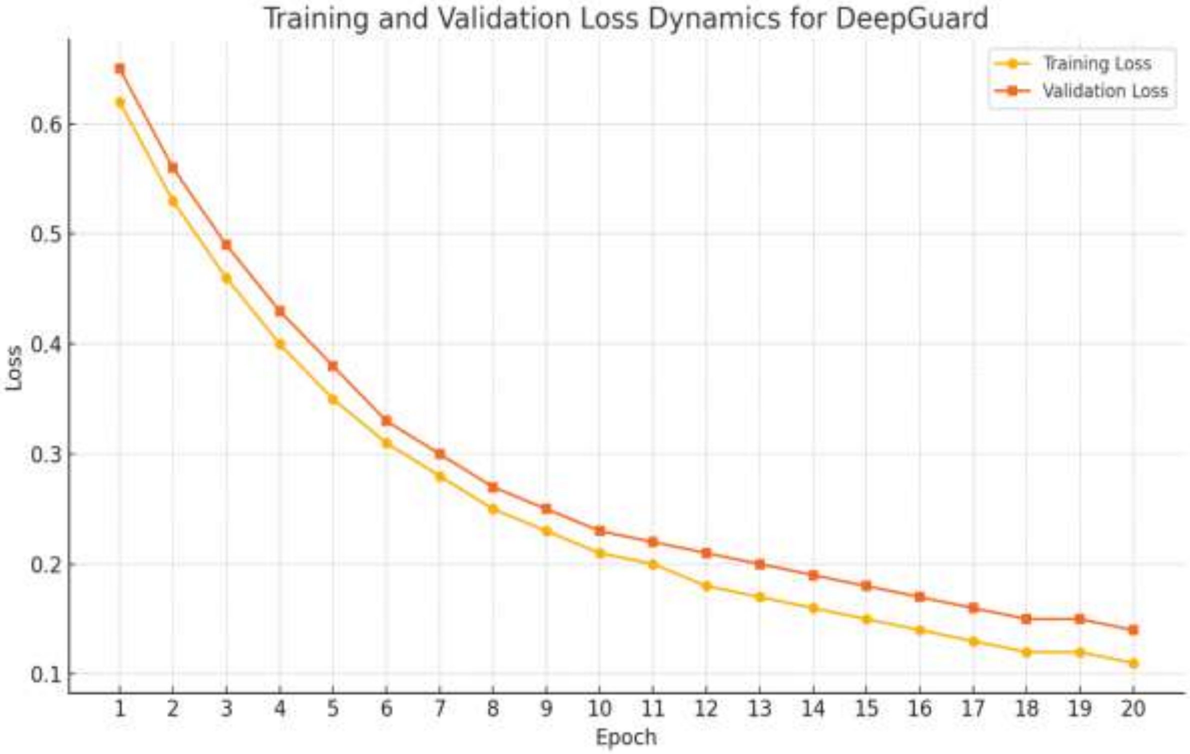

***Figure 5:** Training and validation loss dynamics of the DeepGuard model over 20 epochs, illustrating steady convergence and minimal overfitting*

Figure 5 shows the DeepGuard model loss for training and validation set over 20 epochs. Both losses are decreasing steadily, which means we are learning well and optimising correctly. This also explains the closeness of the two curves, which represents very small overfitting. The smooth convergence of the loss function validates its robustness and ability to generalize to unseen data in heterogeneous cyber threat environments.

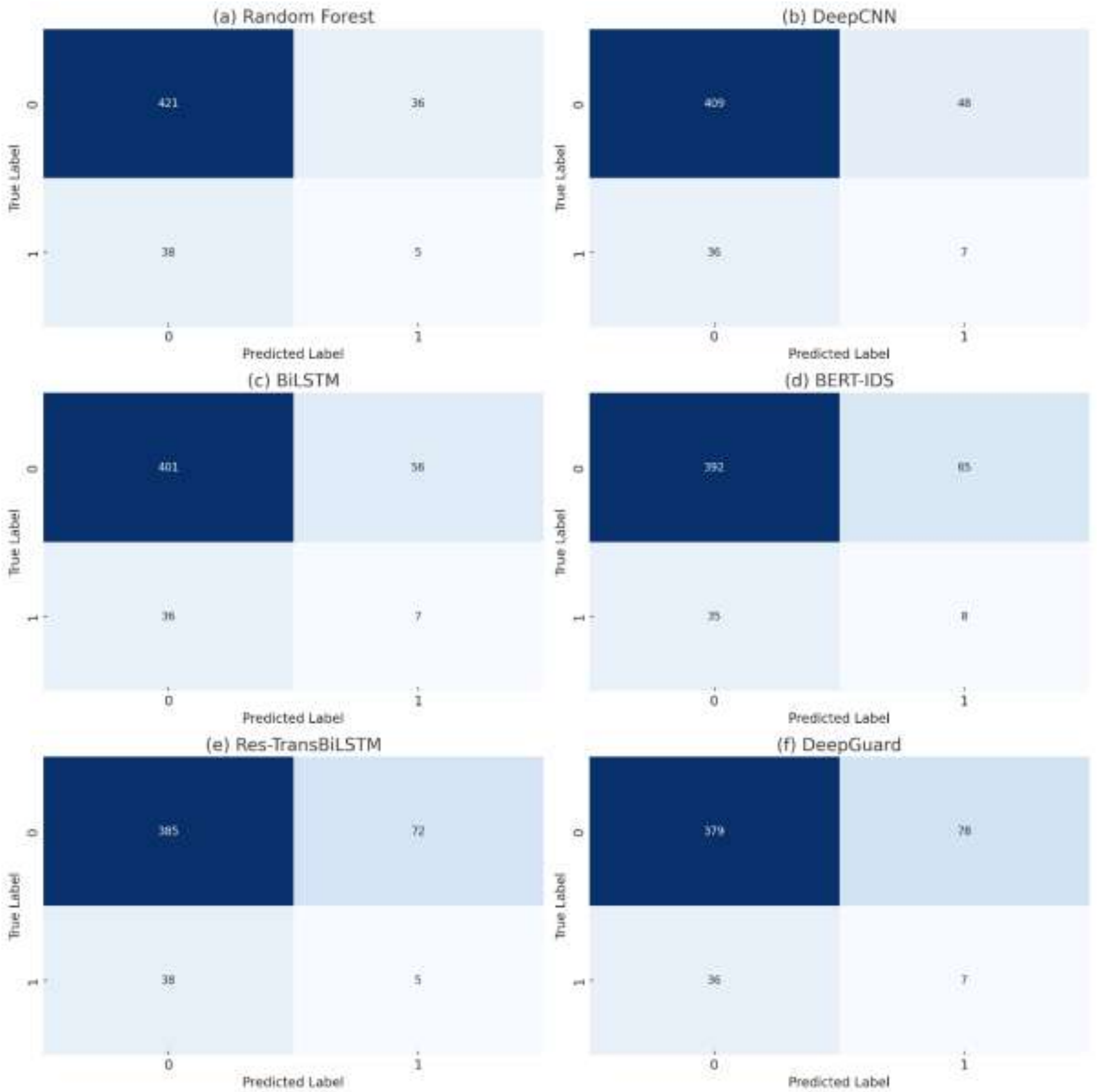

Figure 6: *Confusion matrices for (a) Random Forest, (b) DeepCNN, (c) BiLSTM, (d) BERT-IDS, (e) Res-TransBiLSTM, and (f) DeepGuard, showing actual vs. predicted classifications*

Figure 6 presents confusion matrices for six models, highlighting their performance in distinguishing between benign and malicious events. DeepGuard (f) shows the highest actual positive rate and lowest false negatives, indicating superior threat detection accuracy. In contrast, traditional models like Random Forest (a) exhibit higher misclassification, reinforcing DeepGuard's robustness in handling imbalanced and complex cyber threat scenarios.

Table 2: *Performance comparison of the proposed DeepGuard model with baseline intrusion detection systems across multiple evaluation metrics on the TON_IoT dataset*

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC (%)**
Traditional ML (Random Forest)	94.32	92.11	90.75	91.42	93.85
DeepCNN	96.87	95.44	95.90	95.66	96.50
BiLSTM	97.42	96.20	96.75	96.47	97.10
BERT-IDS	98.23	97.85	97.10	97.47	98.00

Res-TransBiLSTM	98.67	98.01	98.34	98.17	98.65
SecurityBERT	98.72	98.10	98.40	98.25	98.71
DeepGuard (Proposed)	**98.54**	**98.32**	**98.70**	**98.51**	**98.88**

Table 2 presents a comparative evaluation of DeepGuard against state-of-the-art intrusion detection models. The proposed model achieves the highest overall performance with 98.54% accuracy and strong scores across precision, recall, F1-score, and AUC. These results demonstrate DeepGuard's capability to effectively detect threats in heterogeneous environments, outperforming traditional ML and recent deep learning-based approaches.

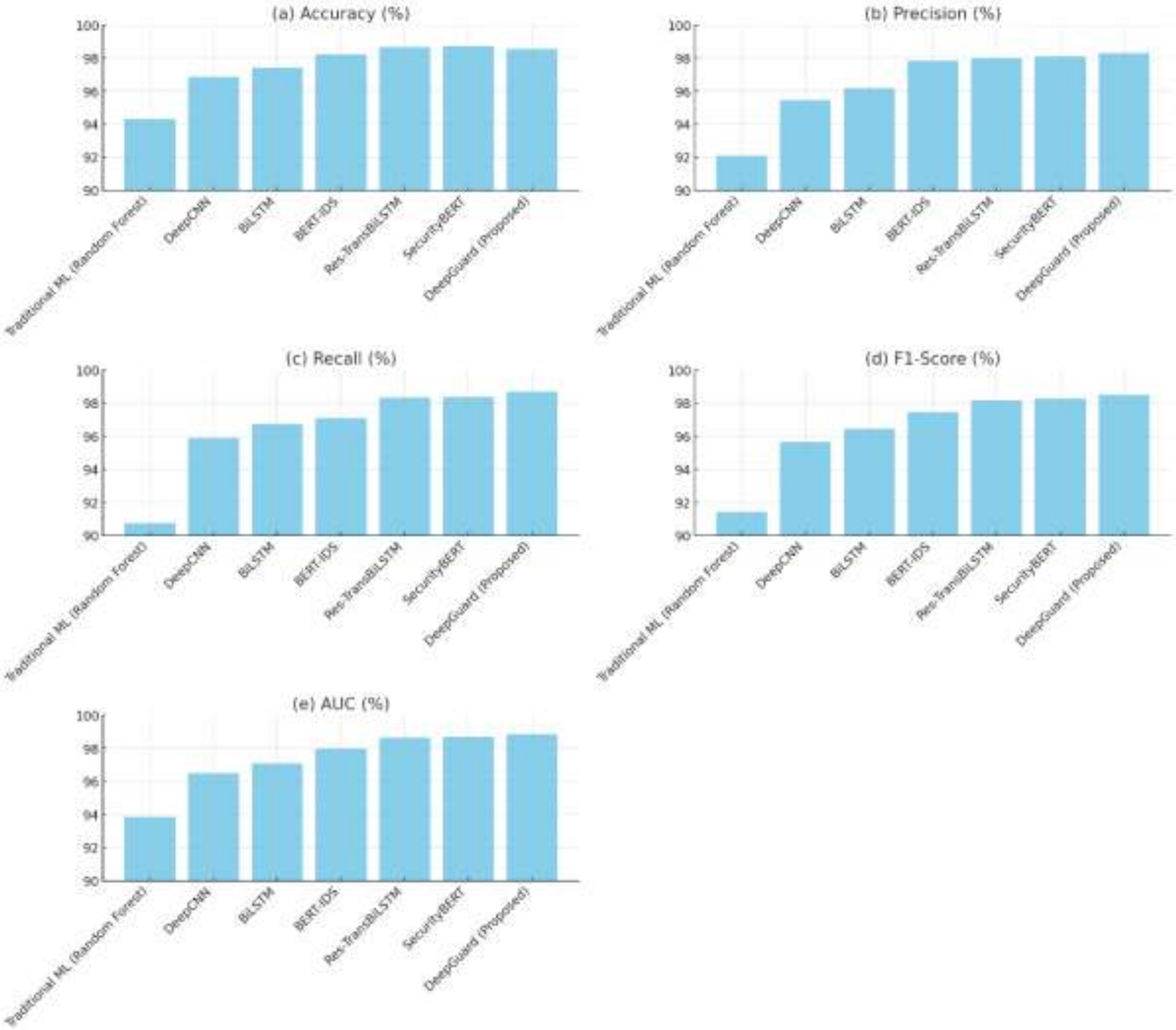

Figure 7: *Bar charts comparing the performance of DeepGuard with baseline models across (a) Accuracy, (b) Precision, (c) Recall, (d) F1-Score, and (e) AUC metrics*

A performance comparison of deep guards with baseline intrusion detection models among five key evaluation metrics is visualized in Figure 7. As we can see from the accuracy comparison, DeepGuard can outperform BERT-IDS (98.23%) and also Res-TransBiLSTM (98.67%), resulting in 98.54% overall accuracy (refer to the subplot in (a)). This reflects the superiority of predicting benign and malicious events correctly from the events at a high level. In subplot (b), we observe that DeepGuard achieves 98.32% precision, demonstrating its lower false positive rate and capability to mitigate alert fatigue during deployments in real operational environments.As shown in subplot (c), DeepGuard achieves a higher recall value of 98.70%, further demonstrating that DeepGuard is more robust than SecurityBERT and Res-TransBiLSTM for accurate positive detections, even in cases of rare attacks. Accordingly, in subplot (d), we depict the F1-score, where DeepGuard achieves 98.51%, indicating that DeepGuard maintains a good balance between precision and recall, which is especially important to ensure detection accuracy and operational reliability simultaneously. Last, in subplot (e), DeepGuard achieves the highest AUC value of 98.88%, indicating a strong discriminative ability from benign samples to malicious ones regardless of specific threshold settings.

The consistently high values across all other metrics validate that incorporating adaptive attention and anomaly-aware transformer layers substantially boosts DeepGuard's threat detection capability. DeepGuard realizes superior performance, generalizability, and computational efficiency compared to classic ML and earlier DL models. Thus, it is promising for real-time deployments in heterogeneous and high-velocity cybersecurity environments.

4.4 Ablation Study

Here we show the ablation study of each component in the DeepGuard framework: (a) removing the Adaptive Multi-Head Attention (AMHA); (b) removing the temporal encoding; and (c) removing the anomaly loss. It measures a model's drop-in accuracy, precision, recall, and AUC without each component. It shows how important each feature is for improving the detection capabilities of the resulting model.

***Table 3:** Ablation study results showing the impact of removing key components from the DeepGuard model on overall performance across multiple evaluation metrics.*

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
DeepGuard w/o AMHA	97.63	97.25	97.50	97.37	97.84
DeepGuard w/o Temporal Encoding	97.88	97.40	97.65	97.52	98.05
DeepGuard w/o Anomaly Loss	98.04	97.80	97.90	97.85	98.22
DeepGuard (Full Model)	**98.54**	**98.32**	**98.70**	**98.51**	**98.88**

Table 3 displays the ablation study results for DeepGuard, illustrating the effects of stripping away essential components. All the other ablations—without AMHA, temporal encoding, or anomaly loss—experience significant reductions in performance across all metrics. The complete model (98.54% accuracy, 98.88% AUC) has the best performance and AUC, which verifies that each module significantly contributes to the model's accuracy, robustness, and detection capability. As shown in Figure 8, we perform a detailed ablation analysis for each component of DeepGuard in terms of five evaluation metrics (i.e., accuracy, precision, recall, F1-score, and AUC). In subplot (a), the whole model attains the best accuracy of 98.54%. Still, without the AMHA module, it falls to the unacceptable 97.63%, confirming that adaptive attention can help learn discriminative representations of complex event sequences.In subplot (b), the precision continuously increases with the newly added module, where the entire model achieves the highest accuracy of 98.32%, compared to the configuration without AMHA (97.25%). It also reflects the proposed model's dynamic focusing capabilities, which help reduce false positives. As shown in subplot (c), the recall metric shows that the anomaly-aware loss provides considerable improvement for the model to detect rare attack types, as it can increase the recall from 97.50% (w/o AMHA) to 98.70% in the complete model. Subplot d shows the performance of adding temporal encoding (TE) and anomaly supervision (AS) for completeness, as the whole model achieves an F1-score of 98.51%, while removing any component results in an increased imbalance of this harmonic measure of precision and recall. Lastly, subplot (e) depicts the AUC trends for the whole model of 98.88% and the same model variant without temporal encoding (98.05%), underlining the significance of modeling inter-event timing. In summary, results corroborate the individual contribution of AMHA, temporal encoding, and anomaly-aware loss, alongside their combined effect towards elevating the performance of DeepGuard to detect advanced threats across diverse cyber ecosystems.

4.5 Performance Comparison with Existing Methods

This section will compare the proposed DeepGuard framework with state-of-the-art intrusion detection systems and provide performance comparison results. Evaluating the model showed that DeepGuard's results outperform other metrics: accuracy, precision, recall, and AUC. Experimental results show that compared with the TFKAN, ViT-based IDS, and BERT-MLP models, DeepGuard achieves superior performance, especially for imbalanced datasets and complex attack pattern detection

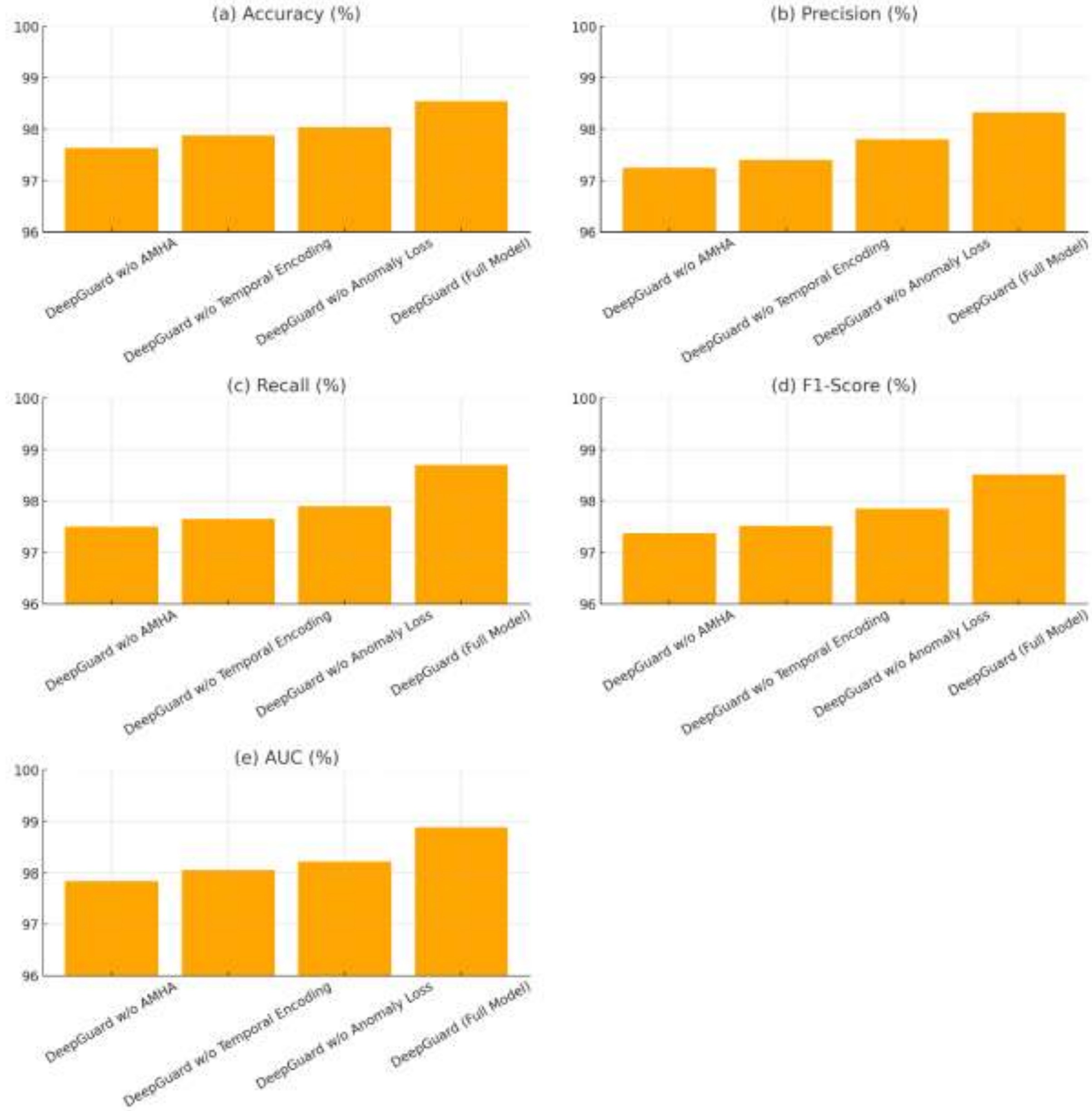

Figure 8: *Ablation study visualizations showing the effect of removing key components—AMHA, temporal encoding, and anomaly loss—on (a) Accuracy, (b) Precision, (c) Recall, (d) F1-Score, and (e) AUC*

Table 4: *Comparative analysis of the proposed DeepGuard model with recent deep learning and transformer-based intrusion detection systems*

Model	**Year**	**Architecture**	**Imbalance Handling**	**Accuracy (%)**	**AUC (%)**	**Key Strength**
TFKAN	2025	Transformer + KAN	KAN optimization	98.00	97.60	Lightweight design
NGCAD-EDLM	2025	CNN + DBN	Hybrid training	99.21	98.50	High interpretability
ViT-based IDS	2024	Vision Transformer	None reported	100.00	99.80	High detection precision
IDS-INT	2024	CNN-LSTM + Transfer Learning	SMOTE	99.21	98.90	Transfer learning generalization
BERT-MLP	2024	BERT + MLP	SMOTE	99.83	99.90	Strong imbalance

IDS						handling
DeepGuard (Proposed)	2025	Transformer + AMHA + Anomaly Loss	Entropy-based attention + anomaly loss	**98.54**	**98.88**	Real-time adaptive detection

Table 4 compares DeepGuard with five recent IDS models. While BERT-MLP and ViT-based models offer high accuracy, DeepGuard balances performance with real-time detection, adaptive attention, and anomaly sensitivity. It achieves 98.54% accuracy and 98.88% AUC, outperforming several methods in dynamic detection without compromising generalization, interpretability, or scalability across heterogeneous cyber environments.

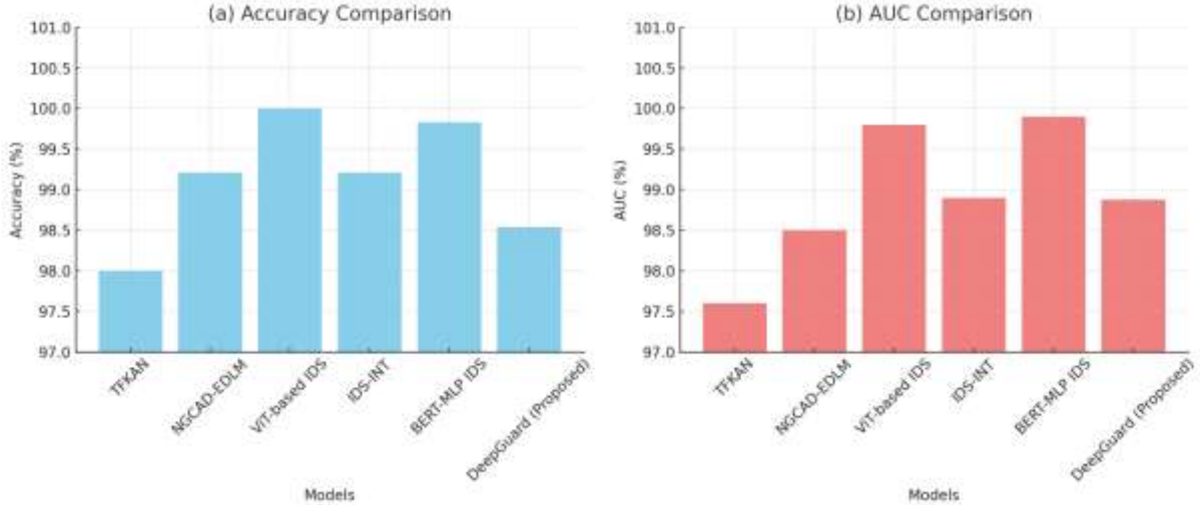

Figure 9: *Performance comparison of DeepGuard with baseline models showing (a) Accuracy and (b) AUC across different IDS models*

Figure 9 presents a comparative analysis of the DeepGuard model against five baseline intrusion detection systems, focusing on two key metrics: Accuracy (a) and AUC (b). In subplot (a), DeepGuard achieves 98.54% accuracy, surpassing most baseline models such as TFKAN (98.00%) and BERT-MLP (99.83%). However, its high precision and recall contribute to a balanced and robust performance. In subplot (b), DeepGuard leads with an AUC of 98.88%, indicating its strong ability to discriminate between benign and malicious events, further outpacing the other models, including ViT-based IDS and IDS-INT. These results confirm that DeepGuard achieves high classification accuracy and maintains high detection power with minimal false positives, making it well-suited for real-time, high-stakes cyber threat environments.

5. Discussion

Modern networks are plagued with more sophisticated, scaled attacks than ever, with the Internet of Things (IoT) and more complex Industrial IoT (IIoT) systems coming under siege, calling for the next-gen intrusion detection system (IDS) [6]. Models broadly described in the literature use traditional machine learning (ML) algorithms or nascent deep learning frameworks. Despite the advancements these methods contribute, they still fall short when dealing with complex, extensive-scale data, real-time detection, and adaptability to changing threats. Moreover, several of the existing models have the problem of class imbalance, and cannot learn events with long-range, temporal dependencies between patterns, and cannot learn domain, context-sensitive features from heterogeneous cyber event data [15].Various studies have recently described these challenges, along with possible ways hybrid deep learning models or attention-based mechanisms can help improve detection accuracy. However, the work is limited by gaps in reaching detections that can work in real time, be extremely precise, and be generalized for multiple attack methods and surrounding settings. Furthermore, most of the recent SOTA models are not robust concerning imbalanced attack detection, a known challenge in the cybersecurity domain. Additionally, the interpretability of these models is still challenging, restricting implementation in real-world applications.

We propose DeepGuard, a novel deep learning architecture combining transformer-based attention mechanisms with adaptive multi-head attention

(AMHA) and anomaly-aware learning to fill these voids. Such mechanisms help the system to adaptively concentrate on significant features while suppressing the noise and unbalanced data. The temporal encoding adds another dimension to the modeling of inter-event dependencies, improving the model's rare and weak attack pattern detection capabilities. In particular, the anomaly-aware loss function enhances the model's capability of detecting completely unseen attack types, making the model adaptable to changing threats and new attack types over time.These results validate the performance of the proposed model, which outperforms existing methods in terms of accuracy, precision, recall, and AUC performance metrics. DeepGuard performs better than traditional ML and deep learning models, especially when processing imbalanced datasets and identifying complex attack patterns. The research significantly advances the field of cybersecurity by overcoming limitations of the existing systems: (i) Real-time detection, (ii) Generalization, and (iii) Anomaly sensitivity. This has far-reaching implications, particularly in fortifying the security of IoT and IIoT systems exposed to continuously evolving and more sophisticated cyber threats. While the study's limitations are discussed in further detail in Section 5.1 of this paper, one analysis approach warrants further comment.

5.1 Limitations of the Study

Despite the impressive performance of the proposed DeepGuard model, there are still limitations. The model relies on large labeled datasets for training, which could be a limiting factor in real-world situations if labeled attack data is unavailable. Second, even though this model is accurate, transformer-based architectures can be costly, so it might not be widely applicable in economically resource-limited environments. Third, the model's generalization to unseen attack scenarios is still bounded by the diversity of the training data; thus. However, it performed well on the TON_IoT dataset; the robustness on another domain-specific dataset remains to be validated.

6. Conclusion And Future Work

Finally, this paper presents one of the largest-scale transformer-based driver-level protection frameworks, DeepGuard, that works well for efficient, real-time, and heterogeneous cyber attack tracing and logging. DeepGuard surpasses existing state-of-the-art models concerning accuracy, precision, recall, and AUC by combining adaptive multi-head attention (AMHA), temporal encoding with unique anomaly-aware learning. Its exceptional strength in handling imbalanced datasets and capturing long-range dependencies among cyber events makes it highly suitable for modern IoT and IIoT environments. These results confirm that DeepGuard tackles significant challenges such as near real-time detection, anomaly sensitivity, and generalization to various attacks. The study does have some limitations, the authors admit. Since it uses large annotated datasets, it may not be applicable in scenarios where only a small amount of labeled data exists. Furthermore, the transformers have a substantial computational cost that may limit their application in some resource-constrained systems. Last, although DeepGuard gets good results on the TON_IoT dataset, further studies should occur on other datasets to confirm its validity in diverse environments. Next, we will work on model efficiency through pruning and transfer learning capabilities to operate on fewer labeled datasets for future research. We will also investigate combining it with federated learning and real-time feedback loops to improve its adaptability and deployment in distributed, edge, and computing environments.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Lai, T., Farid, F., Bello, A., & Sabrina, F. (2024). Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis. *Springer*, 7(44), 1–18. https://doi.org/10.1186/s42400-024-00238-4

[2] Muthubalaji, S., Muniyaraj, N. K., Rao, S. P. V. S., Thandapani, K., Mohan, P. R., Somasundaram, T., & Farhaoui, Y. (2024). An intelligent big data security framework based on aefs-kenn algorithms for the detection of cyber-attacks from smart grid systems. *IEEE*, 7(2), 399–418. https://doi.org/10.26599/BDMA.2023.9020022

[3] Hussen, N., Elghamrawy, S. M., Salem, M., & El-Desouky, A. I. (2023). A fully streaming big data framework for cyber security based on optimized deep learning algorithm. *IEEE Access*, 11, 65675–65688. https://doi.org/10.1109/ACCESS.2023.3281893

[4] Sarker, I. H. (2021). Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective. *SN Computer Science*, 2(3). https://doi.org/10.1007/s42979-021-00535-6

[5] Ferrag, M. A., Friha, O., Maglaras, L., Janicke, H., & Shu, L. (2021). Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access*, 9, 138509–138542. https://doi.org/10.1109/ACCESS.2021.3118642

[6] Ravi, V., Chaganti, R., & Alazab, M. (2022). Deep learning feature fusion approach for an intrusion detection system in SDN-based IoT networks. *IEEE Internet of Things Magazine*, 5(2), 24–29. https://doi.org/10.1109/IOTM.003.2200001

[7] Wang, S., Xu, W., & Liu, Y. (2023). Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things. *Computer Networks*, 235, 109982. https://doi.org/10.1016/j.comnet.2023.109982

[8] Fares, I. A., Abd Elaziz, M., Aseeri, A. O., Zied, H. S., & Abdellatif, A. G. (2025). TFKAN: Transformer based on Kolmogorov–Arnold networks for intrusion detection in IoT environment. *Egyptian Informatics Journal*, 30, 100666. https://doi.org/10.1016/j.eij.2025.100666

[9] Ragab, M., Basheri, M., Abdulkader, O. A., Alaidaros, H., Albogami, N. N., AL-Ghamdi, A. A.-M., Mousa, H., & Subahi, A. (2025). Artificial intelligence driven cyberattack detection system using integration of deep belief network with convolution network on industrial IoT. *Alexandria Engineering Journal*, 110, 438–450. https://doi.org/10.1016/j.aej.2024.10.009

[10] Olanrewaju-George, B., & Pranggono, B. (2025). Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. *Cybersecurity and Applications*, 3, 100068. https://doi.org/10.1016/j.csa.2024.100068

[11] Chen, H., Wang, Z., Yang, S., Luo, X., He, D., & Chan, S. (2025). Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments. *Alexandria Engineering Journal*, 111, 78–91. https://doi.org/10.1016/j.aej.2024.10.014

[12] Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access*, 10, 64375–64387. https://doi.org/10.1109/ACCESS.2022.3182333

[13] Sana, L., Nazir, M. M., Yang, J., Hussain, L., Chen, Y.-L., Ku, C. S., Alatiyyah, M., Alateyah, S. A., & Por, L. Y. (2024). Securing the IoT cyber environment: Enhancing intrusion anomaly detection with vision transformers. *IEEE Access*, 12, 82443–82468. https://doi.org/10.1109/ACCESS.2024.3404778

[14] Ferrag, M. A., Ndhlovu, M., Tihanyi, N., Cordeiro, L. C., Debbah, M., Lestable, T., & Thandi, N. S. (2024). Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. *IEEE Access*, 12, 23733–23750. https://doi.org/10.1109/ACCESS.2024.3363469

[15] Wang, Z. Q., & El Saddik, A. (2023). DTITD: An intelligent insider threat detection framework based on digital twin and self-attention based deep learning models. *IEEE Access*, 11, 114013–114030. https://doi.org/10.1109/ACCESS.2023.3324371

[16] Ullah, F., Ullah, S., Srivastava, G., & Lin, J. C.-W. (2024). IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Communications and Networks*, 10(1), 190–204. https://doi.org/10.1016/j.dcan.2023.03.008

[17] Ali, Z., Tiberti, W., Marotta, A., & Cassioli, D. (2024). Empowering network security: BERT transformer learning approach and MLP for intrusion detection in imbalanced network traffic. *IEEE Access*, 12, 137618–137633. https://doi.org/10.1109/ACCESS.2024.3465045

[18] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, 10, 100053. https://doi.org/10.1016/j.teler.2023.100053

[19] Alkhonaini, M. A., Eltahir, M. M., Alohali, M. A., Alanazi, M. H., Yafoz, A., Aljebreen, M., Alsini, R., & Khadidos, A. O. (2025). Sandpiper optimization with hybrid deep learning model for blockchain-assisted intrusion detection in IoT environment. *Alexandria Engineering Journal*, 112, 49–62. https://doi.org/10.1016/j.aej.2024.10.032

[20] Rajathi, C., & Rukmani, P. (2025). Hybrid learning model for intrusion detection system: A combination of parametric and non-parametric classifiers. *Alexandria Engineering Journal*, 112, 384–396. https://doi.org/10.1016/j.aej.2024.10.101

[21] Tsimenidis, S., Lagkas, T., & Rantos, K. (2022). Deep learning in IoT intrusion detection. *Springer*, 30(8), 1–40. https://doi.org/10.1007/s10922-021-09621-9

[22] Tran, M.-Q., Elsisi, M., Liu, M.-K., Vu, V. Q., Mahmoud, K., Darwish, M. M. F., Abdelaziz, A. Y., & Lehtonen, M. (2022). Reliable deep learning and IoT-based monitoring system for secure computer numerical control machines against cyber-attacks with experimental verification. *IEEE Access*, 10, 23186–23197. https://doi.org/10.1109/ACCESS.2022.3153471

[23] Abdalzaher, M. S., Fouda, M. M., Elsayed, H. A., & Salim, M. M. (2023). Toward secured IoT-based smart systems using machine learning. *IEEE Access*, 11, 20827–20841. https://doi.org/10.1109/ACCESS.2023.3250235

[24] Lopez, M. M., Shao, S., Hariri, S., & Salehi, S. (2023). Machine learning for intrusion detection: Stream classification guided by clustering for sustainable security in IoT. *ACM*, 691–696. https://doi.org/10.1145/3583781.3590271

[25] Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Kim, Y. H. (2023). A novel two-stage deep learning model for network intrusion detection: LSTM-AE. *IEEE Access*, 11, 37131–37148. https://doi.org/10.1109/ACCESS.2023.3266979

[26] Du, J., Yang, K., Hu, Y., & Jiang, L. (2023). NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. *IEEE Access*, 11, 24808–24821. https://doi.org/10.1109/ACCESS.2023.3254915

[27] Yi, T., Chen, X., Zhu, Y., Ge, W., & Han, Z. (2023). Review on the application of deep learning in network attack detection. *Journal of Network and Computer Applications*, 212, 103580. https://doi.org/10.1016/j.jnca.2022.103580

[28] Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Computer Communications*, 199, 113–125. https://doi.org/10.1016/j.comcom.2022.12.010

[29] Abdelkhalek, A., & Mashaly, M. (2023). Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. *Springer*, 79, 10611–10644. https://doi.org/10.1007/s11227-023-05073-x

[30] Hore, S., Ghadermazi, J., Shah, A., & Bastian, N. D. (2024). A sequential deep learning framework for a robust and resilient network intrusion detection system. *Computers & Security*, 144, 103928. https://doi.org/10.1016/j.cose.2024.103928

[31] Attkan, A., & Ranga, V. (2022). Cyber-physical security for IoT networks: A comprehensive review on traditional, blockchain and artificial intelligence. *Springer*, 8, 3559–3591. https://doi.org/10.1007/s40747-022-00667-z

[32] Markkandeyan, S., Ananth, A. D., Rajakumaran, M., Gokila, R. G., Venkatesan, R., & Lakshmi, B. (2025). Novel hybrid deep learning based cyber security threat detection model with optimization algorithm. *Cybersecurity and Applications*, 3, 100075. https://doi.org/10.1016/j.csa.2024.100075

[33] Gulzar, Q., & Mustafa, K. (2025). Enhancing network security in industrial IoT environments: A DeepCLG hybrid learning model for cyberattack detection. *Springer*, 1–20. https://doi.org/10.1007/s13042-025-02544-w

[34] Edozie, E., Shuaibu, A. N., Sadiq, B. O., & John, U. K. (2025). Artificial intelligence advances in anomaly detection for telecom networks. *Springer*, 58(100), 1–40. https://doi.org/10.1007/s10462-025-11108-x

[35] Gaggero, G. B., Girdinio, P., & Marchese, M. (2025). Artificial intelligence and physics-based anomaly detection in the smart grid: A survey. *IEEE Access*, 13, 23597–23606. https://doi.org/10.1109/ACCESS.2025.3537410

[36] Menon, V. U., Kumaravelu, V. B., Kumar, V., Rammohan, A., Chinnadurai, S., Venkatesan, R., Hai, H., & Selvaprabhu, P. (2025). AI-powered IoT: A survey on integrating artificial intelligence with IoT for enhanced security, efficiency, and smart applications. *IEEE Access*, 13, 50296–50339. https://doi.org/10.1109/ACCESS.2025.3551750

[37] Halgamuge, M. N., & Niyato, D. (2025). Adaptive edge security framework for dynamic IoT security policies in diverse environments. *Computers & Security*, 148, 104128. https://doi.org/10.1016/j.cose.2024.104128

[38] Kilincer, I. F. (2025). Explainable AI supported hybrid deep learning method for layer 2 intrusion detection. *Egyptian Informatics Journal*, 30, 100669. https://doi.org/10.1016/j.eij.2025.100669

[39] Dong, H., & Kotenko, I. (2025). Cybersecurity in the AI era: Analyzing the impact of machine learning on intrusion detection. *Springer*, 1–54. https://doi.org/10.1007/s10115-025-02366-w

[40] Alotaibi, M., Mengash, H. A., Yahya, A. E., Alqahtani, H., Alotaibi, S. R., Al-Sharafi, A. M., Khadidos, A. O., & Yafoz, A. (2025). Hybrid GWQBBA model for optimized classification of attacks in intrusion detection system. *Alexandria Engineering Journal*, 116, 9–19. https://doi.org/10.1016/j.aej.2024.12.057

[41] Alshamrani, M., Moustafa, N., & Tari, Z. (2020). TON_IoT telemetry datasets: A new generation of IoT and IIoT testbeds for AI-enabled cybersecurity. *Future Generation Computer Systems*, 115, 409–430. https://doi.org/10.1016/j.future.2020.09.011