JOURNAL OF BASIC SCIENCE AND ENGINEERING

A Hybrid CNN–Transformer Gated Fusion Intrusion Detection System for Edge-Enabled Smart Homes

Khaja Moinuddin[1*], Prabhavathi S[2]

[1, 2]Rao Bahadur Y Mahabaleswarappa Engineering College, Ballari, Karnataka, India-583104

[1,2]Visvesvaraya Technological University, Belagavi, Karnataka, India

DOI: https://doie.org/10.10399/JBSE.2025328313

Abstract: Smart homes rely on interconnected IoT devices, increasing vulnerability to sophisticated cyber attacks. Intrusion Detection Systems (IDS) are essential for safeguarding the IoT ecosystem. Although recent deep learning models have achieved high accuracy in intrusion detection, their substantial computational requirements hinder deployment on resource-constrained IoT devices. To address this challenge, we propose CTGF-IDS (CNN–Transformer Gated Fusion Intrusion Detection System), a lightweight yet powerful intrusion detection framework. CTGF-IDS integrates convolutional layers for efficient local feature extraction with Transformer-based attention mechanisms for modelling long-range dependencies, while a gated fusion module optimizes multi-scale feature integration. This architecture minimizes computational overhead and enhances on-device resource utilization without sacrificing accuracy. CTGF-IDS achieves a substantial reduction in floating-point operations, enabling real-time detection with improved precision, speed, and energy efficiency. Furthermore, knowledge distillation strengthens CTGF-IDS against shifts in traffic distribution, ensuring adaptability in dynamic IoT environments. We evaluate CTGF-IDS using the BoT-IoT and CIC-IDS2017 benchmark datasets, and results demonstrate superior performance compared with existing intrusion detection approaches. The model significantly reduces parameter count, FLOPs, and memory footprint while maintaining high detection accuracy across diverse network traffic patterns. By combining efficiency, robustness, and scalability, CTGF-IDS advances next-generation intelligent IoT security.

Keywords**:** *IOT Security, Intrusion detection, Transformer, Gated fusion. Smart home*

1. Introduction

The Internet of Things (IoT) is expanding at an exceptional pace. Smart devices now constitute a critical component of modern digital infrastructure [1]. By 2025, connected smart devices are projected to exceed 27 billion, representing a doubling compared to statistics reported in 2021 [2]. IoT technologies continue to transform daily life across diverse application domains including smart homes, healthcare, transportation, agriculture, and industrial automation. Each connected device collects, analyzes, and transmits data in real time, thereby enhancing operational efficiency and enabling intelligent decision-making. Automation processes benefit significantly from continuous, data-driven optimization [3].

However, the rapid expansion of IoT has introduced severe cyber security challenges [4]. Security advancements have not kept pace with the growth of connected infrastructures, exposing vulnerabilities across heterogeneous device ecosystems [5]. IoT devices differ widely in hardware configuration, firmware design, and built-in security standards. This heterogeneity complicates the deployment of universal protection mechanisms. Adversaries exploit these inconsistencies through advanced and targeted cyber attacks—including data exfiltration, service disruption, and resource hijacking [6–7]. Threats such as spear phishing, ransomware, and zero-day exploits frequently bypass conventional security architectures with alarming

success. The consequences range from financial loss and operational downtime to threats against critical public infrastructure. These concerns highlight the urgent need for adaptive, scalable IoT defence strategies that ensure confidentiality, integrity, and resilience in distributed environments.

Intrusion Detection Systems (NIDS) have emerged as essential tools for IoT security. NIDS monitor traffic patterns and analyze communication flows in real time to identify anomalies [8]. They detect malicious behavior, support attack mitigation, and provide forensic indicators for threat investigation. These systems enhance network visibility and help administrators assess risk. NIDS solutions are traditionally categorized into misuse-based and anomaly-based models. Misuse-based systems rely on predefined signatures and achieve high accuracy for known threats, but they fail to detect novel or evolving attacks. Anomaly-based systems establish behavioural baselines and detect deviations indicative of malicious activity [9]. Modern anomaly-based NIDS predominantly employ machine learning for automated behaviourmodelling.

Recent advances in deep learning (DL) have demonstrated superior performance for NIDS tasks [10–12], outperforming classical machine learning techniques in feature extraction and classification. However, large DL architectures impose significant computational and memory demands, making on-device deployment difficult for IoT endpoints with constrained resources. To overcome these limitations, many lightweight optimization techniques—such as model pruning [14], quantization [15], and knowledge distillation (KD) [16]—have been proposed. These methods reduce model complexity while maintaining acceptable performance. Lightweight CNN architectures employing depth wise separable convolutions further reduce computational overhead [17]. Although such operations decrease FLOPs, they often increase memory access and inference latency, limiting real-world efficiency.

To address these inefficiencies, we propose CTGF-IDS (CNN–Transformer Gated Fusion Network), a lightweight, adaptable, and robust intrusion detection architecture tailored for resource-constrained IoT environments. Unlike conventional lightweight CNN-based IDS models, CTGF-IDS integrates convolutional layers for efficient spatial feature extraction, Transformer blocks for capturing global dependencies, and a gated fusion mechanism to adaptively combine multi-scale representations. This hybrid design significantly enhances feature expressiveness while maintaining computational efficiency suitable for embedded IoT platforms.

The gated fusion module selectively emphasizes informative feature channels, minimizing redundant computation and enabling low-latency inference. In addition, the interplay between CNN and Transformer components strengthens the model's ability to learn both local and long-range patterns within network traffic. As a result, CTGF-IDS achieves substantial reductions in FLOPs, memory footprint, and parameter count while preserving high detection accuracy, making it ideal for on-device real-time intrusion detection.

To enhance adaptability under dynamic IoT traffic conditions, CTGF-IDS incorporates a knowledge distillation framework. This mechanism improves resilience against domain shifts and distributional variations that frequently occur across heterogeneous IoT deployments. Conventional IDS models assume stable feature distributions, an assumption that fails in practical multi-device environments. Knowledge distillation transfers generalized representations from a high-capacity teacher model to a compact CTGF-IDS student model, enabling strong performance even with limited or shifting target data. This process ensures

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

robustness, cross-domain adaptability, and improved reliability in non-stationary IoT environments.

CTGF-IDS thus combine computational efficiency, adaptive learning, and architectural expressiveness to address modern IoT security challenges. Its hybrid CNN–Transformer design minimizes energy consumption and latency overhead, while the gated fusion module ensures optimized multi-scale representation learning. The integrated KD framework provides continuous adaptation and resilience, ensuring stable performance across diverse deployment scenarios.

To validate performance, we evaluate CTGF-IDS using the widely adopted BoT-IoT and CIC-IDS2017 benchmark datasets. These datasets contain complex, realistic, and varied network traffic patterns representative of real-world IoT scenarios. Our experimental setup simulates heterogeneous configurations and distributional variations. Results confirm that CTGF-IDS outperforms existing lightweight and conventional IDS models, achieving superior accuracy, precision, and robustness. The architecture demonstrates reduced FLOPs, fewer parameters, and a significantly lower memory footprint, while maintaining high detection quality across shifting and noisy traffic patterns. These results establish CTGF-IDS as a scalable and high-performance NIDS solution for next-generation IoT ecosystems.

In summary, our contributions are threefold:

- First, we develop CTGF-IDS, a lightweight hybrid CNN–Transformer architecture with a gated fusion mechanism that balances expressiveness and efficiency for IoT intrusion detection.
- Second, we incorporate a knowledge distillation framework for robust cross-domain adaptation, enabling reliable performance under heterogeneous and dynamic network environments.
- Third, we conduct comprehensive evaluation across multiple IoT intrusion datasets, demonstrating strong computational efficiency, reduced model complexity, and enhanced detection robustness.

Our findings indicate that CTGF-IDS provides secure, efficient, and adaptive protection for IoT environments, achieving an optimal balance between accuracy, model compactness, and operational efficiency. This architecture directly addresses practical IoT security challenges in distributed and dynamic contexts, establishing a foundation for resilient and intelligent next-generation IoT defence systems.

2. Related Works

Several advanced methods have been developed for intrusion detection in IoT networks. This section critically examines both deep learning (DL) and lightweight NIDS techniques. Each approach contributes uniquely toward improving IoT cyber security and reliability.

Deep Learning-Based IDS

Deep learning methods have revolutionized intrusion detection in recent years.Theyeffectively learn complex patterns from raw network data representations [10]-[12].These models extract discriminative features automatically, improving classification accuracy. As a result, DL models now dominate research on network intrusion detection.Their superior pattern recognition ability enables detection of subtle attack signatures.However, their computational demands limit deployment on resource-constrained IoT devices.

Halbouni et al. [20] developed a CNN-based model named CNN-IDS.It was designed to identify malicious network traffic efficiently and accurately. The approach was validated using the CIC-IDS2017 benchmark dataset. It achieved 99.53% accuracy in multiclass classification tasks. The model also recorded a false alarm rate of only 0.12%.For binary detection, accuracy reached 99.56% with a 99.4% detection rate. CNN-IDS employs three convolutional layers and a single fully connected layer .A dropout rate of 0.2 prevents over fitting and stabilizes model training. The model balances accuracy and computational performance effectively. However, it does not optimize model size or temporal traffic features.

Imrana et al. [21] proposed a bidirectional LSTM (BiDLSTM) for intrusion detection. This model processes input sequences in both temporal directions. It captures forward and backward dependencies within network traffic flows. The BiDLSTM achieves high accuracy using the NSL-KDD dataset. It surpasses standard LSTM and other state-of-the-art models in performance. The false alarm rate was reduced significantly compared to baseline techniques. Despite this, the model requires extensive computation and longer training times. Such requirements restrict its deployment on real-time or embedded IoT platforms.

Lazzarini et al. [22] introduced a stacked ensemble of deep learning models. Their system combines four ensemble-based algorithms to form a hybrid detector. This design leverages multiple DL models to strengthen classification performance. It maintains high accuracy with consistently low false alarm rates. Testing across several datasets confirmed its robustness in diverse environments. However, the approach's high computational demands limit use on IoT devices.

Telikani et al. [23] presented EvolCostDeep, a cost-sensitive deep framework. It incorporates fog computing architecture named DeepIDSFog for IIoTnetworks. The method addresses class imbalance through an evolutionary cost optimization process. The JADE algorithm tunes the cost-sensitive loss function dynamically. Stackedauto encoders (SAE) and CNN layers enable hybrid representation learning. Results show F1-scores of 93.8% and 96.6% across industrial IoT datasets. The approach demonstrates effective balancing of accuracy and cost sensitivity. Nonetheless, its complexity limits deployment on small-scale edge systems.

Akuthota and Bhargava [24] proposed a Transformer-based IoT intrusion detection model. This model applies a self-attention mechanism to analyze traffic sequences efficiently. The Transformer captures long-range dependencies within communication patterns. It achieves over 99% accuracy, precision, and recall across two datasets. Performance exceeds that of traditional ML and CNN-based detection techniques. However, deployment feasibility on low-power edge hardware remains untested. Real-time adaptability and latency optimization are yet to be demonstrated.

Zeng et al. [25] addressed challenges of limited labelled training data. They proposed an unsupervised anomaly detection framework using variational autoencoders (VAE).The system integrates adversarial training to generate multivariate time-series data. Particle swarm optimization (PSO) automates hyper parameter tuning and model configuration. This eliminates the need for manual architecture adjustments. The framework achieves high accuracy and generalization across IIoT benchmarks. However, model complexity remains unsuitable for low-resource edge applications.

Lightweight IDS

Recent studies emphasize lightweight NIDS designs for resource-limited environments [26]-[29].The objective is minimizing computational complexity without compromising detection accuracy. Such models enable real-time intrusion detection on IoT and edge devices.

Zhao et al. [30] developed LNN, a lightweight neural network for intrusion detection. It specifically addresses computational challenges in constrained IoT environments. The approach applies principal component analysis (PCA) for dimensionality reduction.Depth wise convolution and channel shuffle operations enhance feature extraction efficiency. Inverse residual blocks further improve representational compactness and speed. Evaluations on two datasets demonstrate strong classification performance. Binary classification accuracies reached 99.9% and 96.8%, respectively. Multiclass accuracies were 96.7% and 96.5% with reduced model size.The model required 84% fewer floating-point operations than conventional CNNs.However, its FLOP count remains relatively high, allowing for further optimization.

Popoola et al. [31] proposed an IoT-oriented NIDS using LSTM auto encoderarchitecture. The method first compresses network data using an auto encoder.The compressed features are subsequently classified using a deep learning model. This hybrid structure reduces dimensionality and improves detection precision. However, the system focuses primarily on data compression rather than model efficiency. Further tuning is needed for accuracy improvement and computational reduction.

He et al. [32] introduced a protocol-level feature grouping strategy for intrusion detection. The approach uses conventional ML algorithms instead of deep learning models. It achieved accuracy exceeding 99.5% with comparable F1-scores across datasets. This framework significantly reduces computation and memory requirements. Feature grouping improves efficiency in both model training and inference stages. However, the method depends heavily on domain-specific feature engineering. Manual feature design limits adaptability across evolving IoT traffic scenarios.

Lu et al. [33] proposed MODEO-CNN for Industrial IoT intrusion detection. The approach employs multi-objective discrete external optimization (MODEO).It combines neural architecture search (NAS) with hyper parameter optimization (HPO).Hybrid encoding mechanisms configure CNN blocks dynamically for performance balance. Evaluations across three datasets confirm strong accuracy and model compactness. Despite superior results, the evolutionary search process is computationally intensive. This reduces the feasibility of deployment in time-sensitive environments.

Li et al. [34] integrated knowledge distillation (KD) into lightweight NIDS.The approach introduces self-supervised contrastive learning for representation extraction. It transfers learned knowledge into a compact CNN through self-distillation. The method addresses both data scarcity and computational limitations. Evaluations on multiple IoT datasets show high accuracy and generalization. The model demonstrates excellent efficiency for on-device intrusion detection.

S. Zhu et.al [35] proposed LKD-STNN, a lightweight knowledge-distilled spatio-temporal neural network. It employs separable convolutions and BiLSTM to capture temporal dependencies. This structure minimizes parameters while maintaining strong learning capacity. An adaptive temperature control function improves KD learning efficiency. Combined loss functions enhance attention to challenging feature patterns. Experiments reveal over 98% accuracy with 99% parameter reduction. The approach offers high suitability for constrained IoT deployments. However, the KD process occasionally introduces minor information

degradation. Real-world adaptability under diverse IoT conditions requires further investigation.

Yang et al. [36] introduced LNet-SKD, a lightweight NIDS for edge devices. The design includes a novel DeepMax block for feature extraction. It integrates depth wise separable convolutions with max-feature-map layers. Self-knowledge distillation compensates for compression-induced performance loss. This ensures robust detection accuracy at reduced computational cost. Experiments across two datasets validate superior efficiency and robustness. The model outperforms several baselines while maintaining minimal parameter overhead.

The authors in [13] extended KD-based CNN models for industrial CPS security. Their lightweight design achieved an 86% computational reduction over baseline systems. Accuracy decreased only marginally by 0.4% on the NSL-KDD dataset. This confirms KD's potential to compress models effectively for edge devices. However, the model's dependency on hyper parameter tuning remains a challenge. Manual adjustments influence performance and require additional training time.

Complex deep learning architectures generally achieve high intrusion detection accuracy. However, their computational intensity restricts practical deployment on IoT hardware. Lightweight models, in contrast, provide better balance between accuracy and cost. They reduce parameter counts, FLOPs, and latency for real-time inference. Nonetheless, lightweight models often face generalization challenges across domains. Models trained on specific datasets may fail under distributional variation. This issue, known as domain shift, limits real-world applicability in IoT networks. Hence, achieving robustness while maintaining efficiency remains a critical research focus.

Lightweight CNN models effectively reduce FLOPs through efficient convolutional operations. Depth wise separable convolutions and grouped filters have demonstrated strong performance; however, these operators often increase memory access, resulting in higher inference latency on resource-constrained IoT devices [18]. Consequently, architecture-level optimization is necessary to reduce overall data movement and improve real-time processing capability.

To address this limitation, the proposed CTGF-IDS (CNN–Transformer Gated Fusion Network) introduces a hybrid optimization strategy that enhances both efficiency and representational power. Instead of relying solely on conventional lightweight convolutions, CTGF-IDS integrates convolutional layers for spatial feature extraction, Transformer blocks for capturing long-range dependencies, and a gated fusion mechanism that adaptively blends multi-scale features. This architecture ensures more expressive representation learning with minimal computational overhead.

Although CTGF-IDS does not employ traditional partial convolution (PConv), it achieves similar efficiency benefits through selective channel attention and gated fusion, which reduce redundant operations and prioritize the most informative features. These mechanisms significantly lower memory movement, accelerate feature extraction, and enhance lightweight model expressiveness. Residual connections further stabilize training and support efficient gradient flow.

To strengthen adaptability across heterogeneous IoT network environments, CTGF-IDS incorporates a knowledge distillation (KD) framework. KD transfers soft-target knowledge from high-capacity teacher models, improving the student model's generalization capability

under shifting traffic distributions. This integration enhances learning stability, mitigates domain drift, and maintains strong detection accuracy even when labeled data are scarce.

Overall, the combination of CNN–Transformer hybrid features, gated fusion, and knowledge distillation enables CTGF-IDS to achieve an optimal balance among adaptability, precision, and on-device efficiency. This hybrid strategy ensures robust generalization, low latency, and high intrusion detection performance in dynamic real-world IoT environments

Table 1: Comparative Analysis of Deep Learning and Lightweight IDS Approaches

Ref	Method	Technology	Advantages	Weaknesses
[13]	Knowledge Distillation + Triplet Convolutional Neural Network (KD-TCNN)	Deep Learning; Lightweight	Provides a compact and accurate IDS for CPS environments; reduces parameters by 86% while maintaining high accuracy.	Though efficient, model tuning is challenging; depends heavily on hyper parameters and requires time-consuming manual optimization.
[20]	CNN	Deep Learning	Achieves 99.55% accuracy with a low false alarm rate (0.12%); enables automatic feature extraction; suitable for large-scale data processing.	May degrade with unseen attack types; focuses primarily on accuracy rather than computational complexity or efficiency.
[21]	Bidirectional LSTM (BiDLSTM)	Deep Learning (Recurrent Networks)	Captures long-term temporal dependencies; improves accuracy; lowers false alarms; detects rare attacks like U2R/R2L.	Computationally demanding; high training time; unsuitable for real-time or resource-constrained IoT applications.
[22]	Stacking Ensemble of DL Models (MLP, DNN, CNN, LSTM)	Deep Learning	Delivers high accuracy and low false positive rate	Ensemble design adds computational complexity; limited scalability for resource-limited IoT devices.
[23]	EvolCostDeep (Stacked Autoencoders + CNN + Cost-Sensitive Learning)	Deep Learning	Addresses data imbalance effectively; integrates cost-sensitive learning using JADE optimization; improves minority attack detection.	Training is resource-heavy due to complex cost optimization; slower convergence and higher computation cost.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

[24]	Transformer-Based IDS with Multi-Head Self-Attention	Deep Learning	Exceeds 99% accuracy; handles long-term dependencies effectively; scalable for large IoT networks.	Requires significant computational resources; lacks validation in constrained IoT edge environments.
[25]	Adversarial Variational Autoencoder (VAE) + PSO	Deep Learning; PSO	Automates hyper parameter search; improves robustness; achieves 97.1% precision and 96% recall; requires no labelled data.	Dependent on optimization phase; costly model pretraining may delay deployment readiness.
[30]	Lightweight Neural Network (LNN) + PCA + Pointwise Convolution	Deep Learning; Lightweight	Attains high classification accuracy with low computational complexity; reduces FLOPs; compact and memory-efficient.	Despite optimization, FLOPs remain relatively high; offers room for further performance refinement.
[31]	LSTM Autoencoder + Deep Bidirectional LSTM	Deep Learning	Reduces data dimensionality; improves generalization; resists overfitting; achieves consistent detection across traffic patterns.	Focuses mainly on dimensionality reduction; lacks emphasis on model optimization and architectural refinement.
[32]	Feature Grouping-Based Lightweight IDS	Machine Learning	Yields >99.5% accuracy; interpretable features; lowers memory and CPU use; supports real-time IoT.	Relies on extensive manual pre-processing; limited adaptability to unseen or evolving attack scenarios.
[34]	Self-Supervised Contrastive Learning + Self-Knowledge Distillation + Depthwise CNN	Deep Learning; Lightweight	Offers a lightweight real-time IDS with high accuracy; enhances feature extraction and generalization; reduces model size.	Incorporates complex model design; training pipeline is computationallyexpensive with higher FLOPs.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

[35]	LKD-STNN: Knowledge Distillation with Depthwise CNN and BiLSTM	Deep Learning; Lightweight	Delivers strong accuracy and compact design; efficient spatio-temporal learning; suitable for IoT edge devices.	Depends on temperature tuning and KD configuration; sensitive to teacher model quality.
[36]	Self-Knowledge Distillation (SKD) + DeepMax Block (DSConv + MFM)	Deep Learning; Lightweight CNN	Achieves >98% accuracy; minimizes parameters by 62%; supports efficient real-time IoT deployment.	Requires precise tuning of hyper parameters; performance may drop with poor initialization.
[33]	MODEO-CNN: Multi-Objective Discrete Extremal Optimization	Deep Learning; Lightweight	Automatically designs CNNs with optimized complexity; performs well on multiple IIoT datasets; balances accuracy and size.	Uses computationally expensive evolutionary search; sensitive to parameter initialization and tuning.

From the literature, DL-based IDSs show unmatched accuracy in threat detection as shown in table 1.However, their deployment in IoT ecosystems is hindered by computational intensity. Lightweight deep models successfully reduce complexity for embedded implementations. These models prioritize FLOPs reduction and energy efficiency during inference.Yet, maintaining detection performance under dynamic network environments remains challenging.Most studies optimize for specific datasets, limiting real-world generalization.Our proposed method addresses these limitations through two integrated solutions.

First, CTGF-IDSemploys partial convolution for reduced computational overhead.It processes only selected input channels for efficient feature learning.This operation minimizes floating-point operations and optimizes spatial representation.Second, knowledge distillation introduces adaptability through cross-domain learning.This framework enhances robustness against dataset bias and distributional drift.Comprehensive evaluations demonstrate its superiority over conventional baselines.The approach achieves balance among accuracy, efficiency, and generalization.Consequently, CTGF-IDSrepresents a scalable solution for modern IoT security. It aligns with emerging requirements of edge-computing-based cyber security architectures.

3. Proposed Approach

This section presents our proposed lightweight intrusion detection model, CTGF-IDS (CNN–Transformer Gated Fusion Network), designed specifically for IoT environments. It also

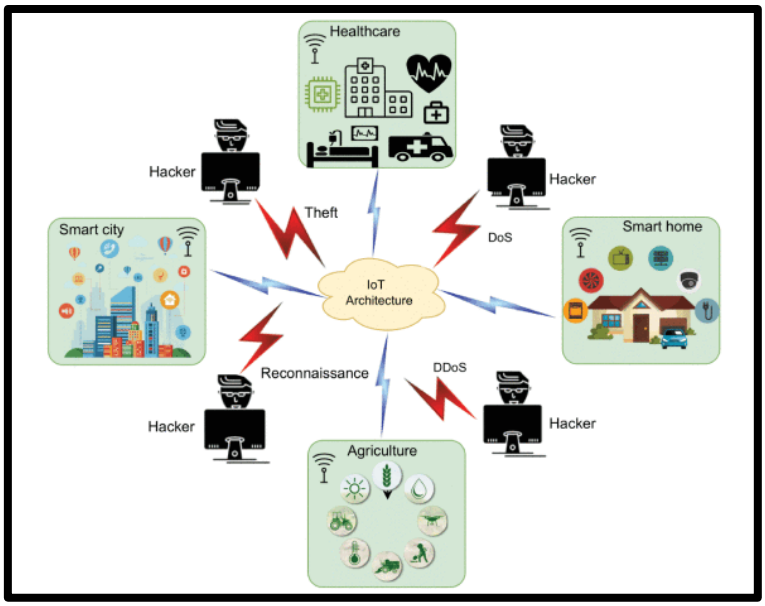

Fig 1: Illustrating the IoT ecosystem and the intrusion attacks landscape.

outlines how knowledge distillation enhances the model's generalization capabilities across diverse network domains. CTGF-IDS combines architectural efficiency with high detection accuracy by integrating the complementary strengths of convolutional neural networks and Transformer-based global attention mechanisms, as illustrated in Fig. 1.

The model achieves an effective balance between predictive precision and computational resource efficiency, enabling deployment on resource-constrained IoT devices.

Unlike previous lightweight IDS architectures that rely on handcrafted or hardware-driven compression techniques, CTGF-IDS employs a modular hybrid design. The system begins with an Expansion Layer that transforms raw network traffic data into a higher-dimensional representation suitable for deep feature extraction. Following this, the architecture integrates CNN blocks for capturing localized spatial patterns and Transformer blocks for modelling long-range dependencies within the traffic sequence. These components are unified within the CTGF Block, which forms the core computational unit of the architecture.

This gating mechanism allows the model to dynamically emphasize the most informative patterns depending on the characteristics of the input traffic sample. Residual connections within the block further stabilize training and maintain continuity of information flow, strengthening deep feature extraction without increasing computational burden.

The combined effect of CNN-based locality modelling, Transformer-based global reasoning, and adaptive fusion produces rich multi-scale feature representations. These representations enable CTGF-IDS to detect both bursty, short-range anomalies and subtle, distributed attack patterns. Since all core operations rely on lightweight convolution and optimized attention mechanisms, the computational footprint remains small, making the model suitable for real-time IoT intrusion detection.

Knowledge distillation further enhances the generalization performance of CTGF-IDS. Through multi-teacher distillation, the student CTGF-IDS model inherits knowledge from multiple teacher networks trained on different source domains. Softened probability distributions and intermediate feature representations guide the student toward domain-invariant feature learning. This process significantly improves robustness against distributional

shifts commonly encountered in heterogeneous IoT deployments. As a result, CTGF-IDS maintains stable performance even when exposed to unseen or evolving traffic conditions.

Together, the CTGF architecture and the knowledge distillation framework deliver an optimal trade-off between speed, accuracy, and adaptability. The model's efficiency stems from its lightweight modular structure, while its detection strength originates from the synergistic integration of CNN and Transformer components. Consequently, CTGF-IDS supports reliable, scalable, and energy-efficient intrusion detection across a wide range of IoT infrastructures.

3.1 Proposed Lightweight CTGF-IDS Model

Figure 2 illustrates the general architecture of our proposed lightweight intrusion detection framework, CTGF-IDS (CNN–Transformer Gated Fusion Network). The model integrates convolutional neural networks (CNNs) and Transformer-based global attention mechanisms within a unified gated fusion architecture. The overall pipeline begins with an expansion layer, followed by a sequence of CNN and Transformer blocks arranged in alternating fashion. These components progressively refine the extracted representations through local feature learning, global dependency modelling, and adaptive fusion. After the final feature extraction stage, a Global Average Pooling (GAP) layer, a dense layer, and a SoftMax classifier generate the final intrusion detection probabilities.

The expansion layer serves as the first stage of the architecture. Its purpose is twofold: (1) to extract preliminary features from raw network traffic sequences, and (2) to expand the input channel dimensionality to a higher-dimensional feature space suitable for subsequent CNN and Transformer operations. Consider an input sequence($x \in R^{L\times C}$) , where L denotes the sequence length (e.g., 36 features) and C=1 represents the initial channel. The expansion layer applies a one-dimensional convolution:

$$F_0 = \text{Conv1D}(x)$$

This operation transforms the input into a feature representation($F_0 \in R^{L\times d}$) , where ddd denotes the expanded channel dimension. This channel expansion increases representational richness and provides sufficient capacity for hybrid local-global feature modeling in deeper layers.

Following channel expansion, the model employs alternating CNN blocks and Transformer blocks, which form the computational backbone of CTGF-IDS. Each CNN block focuses on learning local spatial dependencies, detecting short-range correlations and localized patterns commonly associated with malicious behaviour in network flows. A typical CNN block applies a convolution, normalization, and rectified linear unit (ReLU) activation:

$$F_{\text{CNN}} = \sigma\left(\text{BN}\left(\text{Conv}_{1D}(F_{\text{in}})\right)\right)$$

where σ denotes the ReLU activation. These blocks efficiently capture local variations and high-resolution discriminative patterns.

Complementing the CNNs, the Transformer blocks model long-range temporal and contextual dependencies across network traffic sequences. Unlike CNNs, Transformers utilize multi-head self-attention to compute relationships between all feature positions. Given an input F_{in}, a Transformer block computes:

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

With queries$Q = F_{\text{in}}W_Q,\ K = F_{\text{in}}W_K,\ V = F_{\text{in}}W_V$. The final output is produced through a feed forwardsub layer:

$$F_{\text{Trans}} = \text{FFN}\big(\text{Attention}(Q, K, V)\big)$$

The CTGF Block (CNN–Transformer Gated Fusion Block) is the central innovation of the architecture. It integrates a CNN sub-branch, a Transformer sub-branch, and a gated fusion unit. Given input features F_{in}, the CNN branch computes:

$$F_{\text{local}} = f_{\text{CNN}}(F_{\text{in}})$$

and the Transformer branch computes:

$$F_{\text{global}} = f_{\text{Trans}}(F_{\text{in}})$$

The gated fusion mechanism adaptively merges local and global representations using a learnable gating function:

$$G = \sigma(W_g\,[\,F_{local} \,\|\, F_{global}\,] + b_g)$$

where‖ denotes concatenation, and σ is the sigmoid function. The fused representation is then computed as:

$$F_{\text{fused}} = G \odot F_{\text{local}} + (1 - G) \odot F_{\text{global}}$$

This gate learns how much weight to assign to each branch, dynamically adapting to input characteristics. For instance, attacks with highly localized features benefit from CNN emphasis, whereas globally dispersed signals are better captured through Transformer attention.

Each CTGF block includes a residual connection, enabling stable gradient flow and mitigating vanishing gradient issues:

$$F_{\text{out}} = \text{ReLU}(F_{\text{fused}} + F_{\text{in}})$$

The residual pathway ensures efficient information propagation throughout the network, preserving crucial features across layers.

After passing through stacked CTGF blocks, the feature maps undergo Global Average Pooling (GAP), which aggregates information across temporal dimensions:

$$F_{\text{GAP}}(c) = \frac{1}{L}\sum_{i=1}^{L} F_{\text{in}}(i, c)$$

This operation reduces dimensionality without introducing additional parameters. GAP enables the network to generalize across varied sequence lengths and minimizes over fitting risk.

Subsequently, the pooled features are fed into a dense (fully connected) layer that performs discriminative transformation:

$$F_{\text{FC}} = W_d F_{\text{GAP}} + b_d$$

Finally, a SoftMax activation generates normalized class probabilities:

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

$$P(y = k \mid x) = \frac{e^{F_{FC}(k)}}{\sum_j e^{F_{FC}(j)}}$$

This distribution provides interpretable probabilities for each intrusion category or benign traffic class.

In contrast to earlier lightweight IDS architectures that rely heavily on specialized convolutional operators (such as depth wise separable convolutions or partial convolutions), CTGF-IDS achieves efficiency and accuracy through its hybrid local-global modelling and adaptive fusion strategy. The gated fusion mechanism eliminates the need for handcrafted convolutional optimizations, while the Transformer blocks allow the network to handle complex cross-feature dependencies. Moreover, the use of residual connections and reduced-parameter CNN layers ensures computational efficiency suitable for edge and IoT deployment.

Overall, CTGF-IDS achieves a balanced trade-off between representational richness and computational affordability. By combining CNN-based local extraction, Transformer-based global reasoning, and an intelligent gating mechanism, the architecture provides high detection performance across diverse IoT environments. Its modular, scalable design makes it suitable for both centralized and on-device intrusion detection scenarios, addressing the growing security challenges of the IoT ecosystem.

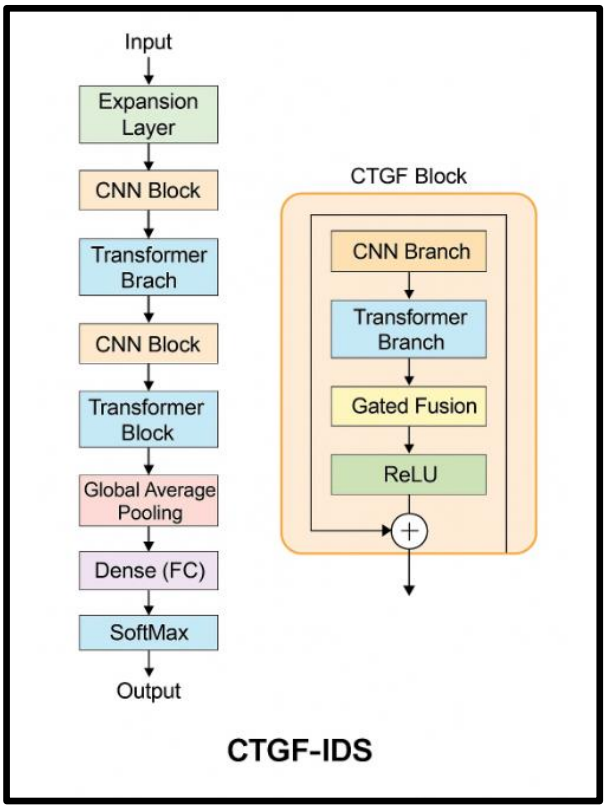

Fig 2: CTGF-IDS model architecture.

3.2 Enhancing Model Generalizability through Knowledge Distillation (KD)

Knowledge Distillation (KD) is a powerful model compression and transfer-learning strategy in which informative knowledge learned by a high-capacity teacher model is transferred into a more compact and computationally efficient student model. The primary goal is to enable the student to approximate the teacher's predictive behaviour while operating with substantially

lower computational overhead [16]. This approach helps the student model acquire high-level semantic understanding without requiring the extensive complexity typical of deep and resource-intensive architectures. As a result, KD improves performance while preserving the compactness, energy efficiency, and fast inference essential for real-time IoT intrusion detection.

In recent years, KD has gained prominence in deep learning research due to its versatility in enhancing generalization performance, compressing large models, and mitigating over fitting. Within the context of intrusion detection, KD plays a critical role in addressing one of the most challenging issues in real-world IoT deployments: domain shift. Traditional intrusion detection systems implicitly assume that training and testing data originate from identical or near-homogeneous distributions. However, IoT networks are inherently dynamic, and their traffic characteristics often evolve over time due to changes in device composition, communication protocols, and environmental conditions. These domain shifts result in altered statistical distributions—degrading model accuracy, destabilizing predictions, and affecting long-term reliability.

To address this challenge, CTGF-IDS integrates Knowledge Distillation as a core component of its training framework. The hybrid CNN–Transformer–Gated Fusion architecture of CTGF-IDS makes it particularly well-suited for KD-based learning. The CNN branches efficiently capture localized spatial features, while the Transformer branches extract global correlations across feature dimensions. The gated fusion mechanism adaptively balances these complementary representations. When KD guidance is added, CTGF-IDS benefits from teacher-supervised refinement at both the local and global levels, enabling the student model to better learn domain-invariant representations.

Unlike traditional ensembling approaches—which improve prediction robustness by aggregating multiple independent model outputs—CTGF-IDS uses KD to compress ensemble knowledge into a single, lightweight student model. Ensemble-based inference is impractical for IoT deployments because it requires simultaneous execution of large models, resulting in increased latency, memory consumption, and energy usage. KD avoids this limitation by transferring the representational diversity of multiple teachers to a single efficient student, significantly reducing inference cost while retaining ensemble-level performance [38], [39]. This enhancement makes CTGF-IDS well aligned with real-time operational constraints in resource-limited IoT environments.

Fine-tuning represents another common strategy for adapting pre-trained models to new data domains. However, fine-tuning is limited by reliance on a single teacher or a narrow dataset. This constrains generalization when models encounter traffic distributions unseen during training. In contrast, KD supports multi-teacher learning, where each teacher is trained on a distinct source domain. Such multi-source knowledge enables richer semantic coverage and diversifies the feature representations available for student learning. As a result, the student model becomes more resilient to adversarial variations, heterogeneous device patterns, and unseen distribution shifts.

Building upon prior KD formulations presented in [38] and [19], our approach extends the methodology to optimize robustness specifically within dynamic IoT intrusion detection environments. In CTGF-IDS, multiple teacher models are trained independently using heterogeneous network traffic sources, each characterized by different temporal, structural, or behavioural properties. The collective knowledge from these teachers forms a high-level supervisory signal that guides the training of the CTGF-IDS student model (Fig. 4). The soft

probabilities and intermediate representations output by the teachers provide nuanced information beyond hard labels, reflecting finer class relationships that are essential for accurate decision-making under distribution variability.

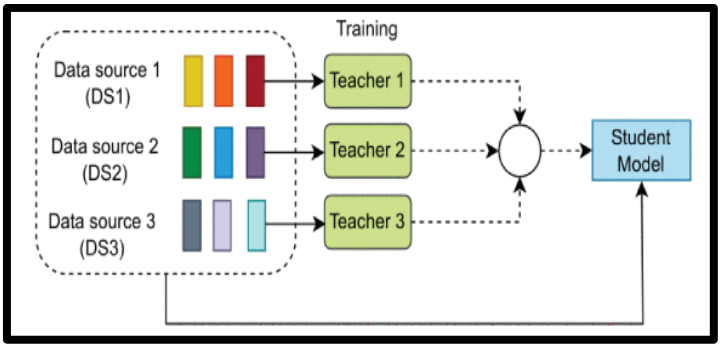

Fig 3: Enhancing model robustness via KD.

Let the teacher ensemble consist of T models, each producing a soft prediction distribution pt(y|x) for input sample x. The aggregated teacher knowledge is expressed as the average softened output:

$$p_{\text{teacher}}(y \mid x) = \frac{1}{T}\sum_{t=1}^{T} \text{Softmax}\left(\frac{z_t}{\tau}\right)$$

wherez_t denotes the teacher logits, and τ is the distillation temperature that smoothes the probability distribution. The CTGF-IDS student output is defined as:

$$p_{\text{student}}(y \mid x) = \text{Softmax}\left(\frac{z_s}{\tau}\right)$$

Traditional KD relies solely on distillation at the output logits. However, because CTGF-IDS contains hybrid CNN and Transformer branches, additional performance gains can be achieved by distilling intermediate representations. Let the teacher ensemble contain intermediate features from layer l, denoted:

$$H_t^{(l)} = f_t^{(l)}(x)$$

and the corresponding student representation:

$$H_s^{(l)} = f_s^{(l)}(x)$$

To preserve structural similarity between teacher and student features across both local (CNN) and global (Transformer) streams, we incorporate an Intermediate Feature Matching Loss defined as:

$$L_{\text{IFM}}^{(l)} = \frac{1}{2}||\phi\left(H_t^{(l)}\right) - H_s^{(l)}||_2^2$$

where $\phi(\cdot)$ is a lightweight projection (e.g., 1×1 convolution or linear transform) used to match channel dimensions when necessary.

To unify multiple layers, the overall intermediate-layer loss is:

$$L_{\text{IFM}} = \sum_{l \in \mathcal{L}} \beta_l \, L_{\text{IFM}}^{(l)}$$

with β_l controlling the contribution of each distilled layer.

Training minimizes the combined KD loss:

$$L_{\text{KD}} = \tau^2 \cdot \text{KL}(p_{\text{teacher}} \mid p_{\text{student}})$$

alongside the standard cross-entropy loss:

$$L_{\text{CE}} = -\sum_{c} y_c \, \log\big(p_{\text{student}}(y = c)\big)$$

yielding the total training objective:

$$L_{\text{total}} = \alpha \, L_{\text{KD}} + (1 - \alpha) \, L_{\text{CE}}$$

where α balances the influence of teacher guidance and ground-truth labels.

Through this multi-teacher KD strategy, the student version of CTGF-IDS learns more stable and domain-generalizable features, benefiting from diverse local-global patterns originally captured by the teachers. The CNN layers learn to extract domain-invariant local patterns guided by teacher supervision, while the Transformer layers gain globally consistent relational embeddings that remain robust under changing traffic conditions. The gated fusion mechanism further ensures that the student effectively combines these complementary signals.

Consequently, CTGF-IDS achieves strong generalization performance even when deployed in heterogeneous, evolving IoT networks. By leveraging KD, the model becomes resilient to domain shifts, maintains stable predictions under traffic variability, and ensures high detection performance on resource-constrained IoTdevicesall while preserving computational efficiency. The algorithm in Table 2 presents the whole training procedure.

Algorithm 1. Multi-Source Knowledge Distillation for CTGF-IDS

Inputs:

```
   T = {T1, T2, …, TN}          // Set of N teacher models
   S                            // CTGF-IDS student model
D_source = {D1, D2, …, DN}  // Source-domain datasets for teachers
D_target                     // Target-domain training dataset for student
τ                            // Temperature for softening logits
α∈ [0,1]                     // Balancing coefficient for KD loss
Output:    Trained student model S
1:  // -------- Teacher Training Phase --------
2:  for each teacher Ti in T do
3:      Train Ti on its corresponding dataset Di
4:  end for
5:  // -------- Student Training Phase --------
6:  for each training batch (x, y) in D_target do
7:      // Teacher soft predictions
8:      for each teacher Ti in T do
9:          z_ti ← Ti(x)                 // Teacher logits
10:         p_ti ← Softmax(z_ti / τ)     // Softened probabilities
```

11: end for
12: p_teacher ← (1/N) * Σ_ip_ti // Aggregated teacher knowledge
13: // Student predictions
14: z_s ← S(x) // Student logits
15: p_student ← Softmax(z_s / τ)
16: // Loss terms
17: L_KD ← τ^2 * KL(p_teacher || p_student)
18: L_CE ←CrossEntropy(y, Softmax(z_s))
19: // Total loss
20: L_total ← α * L_KD + (1 − α) * L_CE
21: BackpropagateL_total and update S's parameters
22: end for
23: return S

4. Experimental Setup

This section details experiments conducted to assess the proposed CTGF-IDS model.The objective is to evaluate classification accuracy, efficiency, and generalization performance.We assess the lightweight design and knowledge distillation effectiveness comprehensively.Additionally, the datasets and evaluation metrics used are clearly described.

4.1 Evaluation of CTGF-IDS Classification and Lightweight Design

Binary and multiclass classification experiments are performed for comprehensive evaluation.We benchmark CTGF-IDS against established lightweight deep learning architectures.Standard evaluation metrics defined in Section 4.7 are used for assessment.For binary classification, all attack categories are combined into one.This converts the intrusion detection task into a binary anomaly detection problem.Datasets introduced in Section 4.2 are utilized for experimental validation.Performance comparisons highlight both accuracy and computational efficiency improvements.Results demonstrate CTGF-IDS maintain precision while reducing processing complexity significantly.

4.2 Evaluation of CTGF-IDS Generalization via Knowledge Distillation

We evaluate generalization improvement through knowledge distillation-based learning.Cross-domain and few-shot adaptation experiments validate model robustness across domains.These experiments employ inter-dataset evaluation using different network traffic sources.Each dataset shares identical feature structures to enable fair comparison.However, existing NIDS datasets vary in format and feature composition.To ensure consistency, we adopt unified datasets proposed in [40].These datasets are transformed into NetFlow-based representations with consistent attributes. The NetFlow versions are derived from raw PCAP files of UNSW-NB15 [41], ToN-IoT [42],BoT-IoT [43],CICIDS2017 [44].All datasets originate from the same research group with distinct configurations.Differences in network setups represent diverse domains and deployment conditions.Such diversity is ideal for testing cross-domain and few-shot adaptability.

We focus on attack classes common to all three datasets.Table 2 presents the final composition of the selected attack classes.Significant class imbalance was observed across the datasets. To address this imbalance, data resampling techniques were applied systematically.We used

random under sampling and Synthetic Minority Over-sampling Technique (SMOTE).These pre-processing steps ensured balanced training and stable evaluation outcomes.The combined framework validates robustness, scalability, and adaptability CTGF-IDS.

Table2: Distribution of selected attack classes

Dataset	Benign Samples	DoS Attacks	Reconnaissance Attacks
ToN-IoT	60,99,469	7,12,609	37,81,419
BoT-IoT	1,35,037	1,66,73,183	26,20,999
UNSW-NB15	22,95,222	5,794	12,779
CIC-IDS2017	22,70,397	2,31,073	1,58,930

This section outlines research questions formulated to evaluate our KD approach. The goal is to determine its effectiveness in improving generalization and adaptability.

RQ1: Does the distillation method enhance student model generalization in unseen domains?
RQ2: How effectively can the student model adapt using few-shot samples?

To address RQ1, we conduct extensive cross-domain evaluation experiments using the complete workflow illustrated in Figure 4. This evaluation follows the multi-teacher knowledge distillation framework described previously in Section 3.2. Within this setup, our CTGF-IDS serves as the student model. Two high-capacity teacher models are first trained independently using source-domain datasets Ds1 and Ds2. These teachers capture heterogeneous intrusion patterns, each learning domain-specific local and global representations. Through the distillation process, the teachers transfer their softened output probabilities and intermediate-layer knowledge to CTGF-IDS.

The objective of this training strategy is to enhance robustness under distributional shift, a critical challenge in practical IoT environments. Because CTGF-IDS includes CNN branches, Transformer branches, and a gated fusion mechanism, it is highly receptive to distilled multi-domain knowledge. CNN sub-modules benefit from teacher guidance for local feature extraction, Transformer sub-modules acquire globally stable relationships, and the gated fusion unit learns how to adaptively combine these representations across varying traffic characteristics. This enables the distilled CTGF-IDS model to maintain predictive consistency even when deployed on unseen domains.

To quantify the impact of distillation, we compare CTGF-IDS performance with and without KD integration. Metrics such as accuracy, precision, recall, and F1-score are analyzed to identify improvements gained through cross-domain knowledge transfer. Because CTGF-IDS is naturally designed to fuse local and global information, the KD-enhanced version demonstrates higher robustness to distributional shift and better generalization across new traffic regimes. The evaluation ultimately focuses on how CTGF-IDS behave when tested on previously unseen target domains, enabling us to directly assess improvements in domain robustness.

To address RQ2, we conduct few-shot domain adaptation experiments, designed to evaluate the adaptability of CTGF-IDS when only a limited number of labelled samples are available from the target domain. This scenario reflects real-world intrusion detection constraints, where annotated network traffic is scarce, costly to obtain, and difficult to label due to privacy considerations and rapid evolution of attack patterns.

Few-shot learning is therefore essential for realistic model deployment. In these experiments, CTGF-IDS receive a very small set of labelled samples from the target domain, drawn according to the configurations listed in Table 3. We incrementally increase the number of labelled samples (e.g., 10, 20, 50 instances) to study how quickly CTGF-IDS performance improves. Stratified sampling ensures balanced class representation; for example, a 10-sample configuration includes 5 benign and 5 attack samples.

This prevents label imbalance and yields a fair comparison across varying sample sizes. These experiments are particularly informative for CTGF-IDS because of its hybrid local–global architecture. With only a handful of labelled samples, CNN layers learn to refine low-level intrusion patterns, Transformer layers adapt global dependencies to the new domain, and the gated fusion component selectively reweights these signals to produce

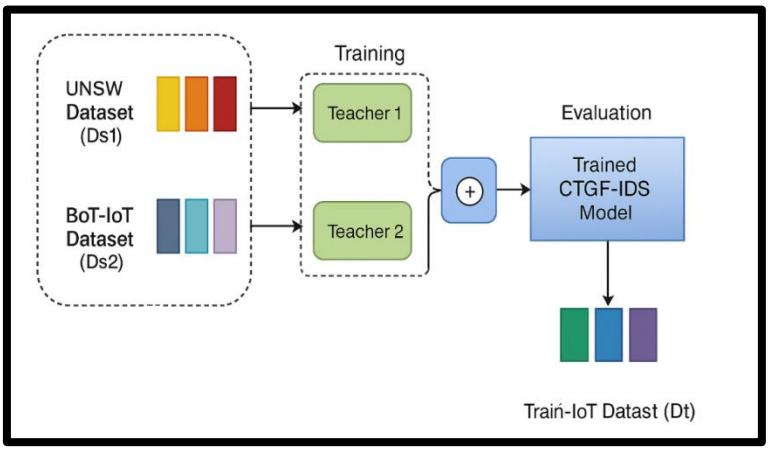

Fig 4: Cross-domain evaluation procedure.

stable predictions. When enhanced by KD, the model starts with a strong domain-generalizable initialization, enabling it to adapt more efficiently under few-shot conditions.

Overall, these few-shot evaluations highlight the adaptability, efficiency, and practical deployability of the distilled CTGF-IDS model. The results demonstrate that CTGF-IDS can rapidly align with newly emerging or previously unseen intrusion behaviours with minimal labelled data. Consequently, our findings validate the effectiveness of the KD-enhanced CTGF-IDS framework in resource-constrained, data-scarce, and continuously evolving IoT environments.

Table 3: Few-Shot Adaptation Results on Target Datasets

Samples in Target Dataset	BoT-IoT (with KD)	BoT-IoT (without KD)	ToN-IoT (with KD)	ToN-IoT (without KD)	UNSW (with KD)	UNSW (without KD)
5	56	43	52	42	48	40
10	67	52	64	46	59	47
20	79	62	73	54	71	60
50	86	73	83	67	82	70
100	90	78	89	80	92	80

4.3 Datasets

This section describes the datasets used for evaluating our proposed framework.Each dataset supports assessing model performance, robustness, and generalization across domains.We selected four widely recognized intrusion detection datasets for experimentation.

4.3.1 BoT-IoT Dataset

The BoT-IoT dataset is a prominent IoT intrusion detection benchmark [43].It was developed by the Cyber Range Lab, University of New South Wales (UNSW), Canberra.The dataset provides realistic IoT network traffic for evaluating intrusion detection systems.It includes both benign and malicious traffic covering diverse IoT attack scenarios. Attack types include DoS, DDoS, reconnaissance, and data theft activities.This dataset is comprehensive, enabling effective evaluation of lightweight IDS models.Table 4 presents the detailed distribution of attack classes in BoT-IoT. Its large volume and diversity make it ideal for deep learning evaluation

Table 4: Class distribution of the BoT-IoT dataset

Category	Train	Test
Benign	381	96
DDoS	11,54,299	3,85,325
DoS	13,20,208	3,30,052
Reconnaissance	**72866**	**18216**
Theft	63	16
Total	25,47,817	7,33,705

4.3.2 CIC-IDS2017 Dataset

The CIC-IDS2017 dataset is another widely adopted NIDS benchmark [44].It was generated by the Canadian Institute for Cyber security (CIC) at UNB.This dataset improves upon earlier ones by simulating modern attack environments.It represents normal and malicious network activities observed in real business contexts.Traffic was captured over five consecutive working days under controlled conditions.The dataset contains realistic modern attacks such as DDoS, infiltration, and brute-force..Table 5 provides the detailed attack distribution of the CIC-IDS2017 dataset.

Table 5: Class distribution of the CIC-IDS2017 dataset

Category	Train	Test
Benign	3,29,959	1,09,724
DoS Hulk	1,72,318	57,806
DoS Goldeneye	**7739**	**2554**
DoS Slowloris	4,358	1,438
DoS Slow httptest	4,170	1,329
Heartbleed	**9**	**2**
Total	**5,18,553**	**1,72,853**

4.3.3 ToN-IoT Dataset

The ToN-IoT dataset was also created by the UNSW Cyber Range Lab [45].It aims to support cybersecurity research across IoT and IIoT environments. The dataset includes heterogeneous data collected from multiple IoT-based sources. Data sources include network traffic, telemetry,

and operating system event logs. ToN-IoT covers nine distinct cyberattack types, representing realistic network conditions [42].We use the NetFlow-based version standardized by the authors of [40].This transformation ensures consistent feature structures across datasets for fair evaluation.The dataset's diversity supports robust cross-domain and few-shot experiments.

4.3.4 UNSW-NB15 Dataset

The UNSW-NB15 dataset is widely recognized for modern NIDS evaluation [41].It was developed by the same UNSW Cyber Range Lab team.The dataset provides realistic representations of recent network traffic and attack behavior. It includes normal and malicious samples generated from modern cybersecurity environments. Attacks include fuzzers, exploits, reconnaissance, DoS, and shellcode-based intrusions.We employ the NetFlow-based format published by the authors of [40].This version ensures compatibility with other datasets used in our experiments. Its diversity supports training models capable of detecting a broad attack spectrum.

4.4 Dataset Pre-processing

Data preprocessing ensures high-quality input for model training and evaluation.This stage involves cleaning, organizing, and normalizing data before model training. A structured preprocessing pipeline improves data reliability and model performance. Our preprocessing steps include feature selection, data cleaning, and normalization.

4.4.1 Feature Selection

Feature selection is crucial for optimizing lightweight intrusion detection models.It enhances accuracy, reduces redundancy, and minimizes computational overhead.We use Random Forest-based feature importance to identify relevant features.This method ensures that only significant features contribute to classification.Additionally, correlation coefficient analysis removes redundant or highly correlated attributes.These combined methods improve interpretability and prevent over fitting during training.Using this approach, we reduced BoT-IoT features from 63 to 8.Similarly, CIC-IDS2017 features were reduced from 76 to 8.This reduction improves efficiency without sacrificing model performance.

4.4.2 Data Cleaning

Data cleaning eliminates errors, missing values, and duplicate entries.It ensures that no data instance is overrepresented in the training process.Non-numeric attributes are transformed using one-hot encoding for compatibility.This process provides consistent numeric inputs for deep learning architectures.The resulting data ensures unbiased and accurate model evaluation.

4.4.3 Data Normalization

Data normalization plays a critical role in stabilizing input feature distributions during the training of CTGF-IDS. As CTGF-IDS integrates convolutional layers, Transformer units, and a gated fusion mechanism, maintaining consistent feature scales is essential for efficient gradient flow and stable attention computations. To achieve this, we employ a hybrid normalization strategy combining MinMax normalization and log-normal normalization [34]. This combination mitigates MinMax sensitivity to extreme feature values while preserving relative magnitude differences for Transformer-based self-attention.

After normalization, we reshape each sample into a 1D sequential format, ensuring compatibility with the CTGF-IDS architecture shown in Figure 2. This structure aligns with the model's design, where CNN branches expect 1D spatial input and Transformer branches process temporal sequences. Uniformly scaled and properly shaped input data enhance model convergence stability, detection precision, and real-time adaptability, especially under dynamic IoT traffic distributions.

4.5 Implementation Details

Table 6 summarizes the architectural configuration of the proposed CTGF-IDS (CNN–Transformer Gated Fusion Network). All hyper parameters are empirically optimized to maximize intrusion detection accuracy while maintaining a lightweight design suitable for edge and IoT deployments. Unlike CTGF-IDS, which relied on partial convolution and channel shuffling, CTGF-IDS uses a hybrid local–global feature extraction pipeline enhanced through gated fusion.

The architecture begins with an Expansion Layer, implemented as a 1D convolution that increases channel dimensionality and extracts initial low-level features. This expanded representation acts as the shared input for the subsequent CNN Block, Transformer Block, and CTGF Block components. Filter sizes are consistently set to 3 for all convolutional operations, preserving spatial alignment while maintaining computational efficiency.

The CNN Blocks employ lightweight 1D convolutions followed by batch normalization and ReLU activations. These layers focus on capturing short-range spatial dependencies characteristic of many IoT attack patterns. Kernel sizes of 3 preserve temporal resolution, ensuring that feature map dimensions remain stable without the need for reshaping.

The Transformer Blocks incorporate multi-head self-attention and feedforward layers. These modules learn long-range interactions between features, which enhances the model's ability to detect complex or slowly evolving intrusions. Key parameters include the number of attention heads, hidden dimensionality, and the size of the feedforward network. These settings are tuned to balance expressiveness with memory efficiency. The CTGF Block represents the core architectural unit. The gating mechanism includes learnable parameters that adaptively combine local CNN-based cues with global Transformer-derived patterns. A sigmoid gate determines the contribution of each branch, enabling flexible representation merging under varying traffic conditions. Residual connections around each CTGF block further stabilize training and enhance feature continuity.

The number of CTGF blocks is fixed at two to maintain architectural simplicity while providing adequate depth for hierarchical feature learning. This configuration ensures effective internal fusion without introducing excessive computational overhead

Table 6: CTGF-IDS -IDS architectural details

Layer / Component	Configuration
Expansion Layer	Conv1D(filters=16, kernel=3)
CNN Block	Conv1D → BN → ReLU, kernel=3
Transformer Block	Multi-Head Attention + FFN

CTGF Block	CNN Branch + Transformer Branch + Gated Fusion + Residual
Global Avg Pooling	1D GAP
Dense Layer	Fully Connected
Output Layer	SoftMax

All teacher models used in the KD framework share the same CTGF architecture, ensuring consistent representational structure during knowledge transfer. The distillation coefficient is set to α=0.1, balancing the influence of teacher-provided soft targets with ground-truth supervision. The temperature parameter is fixed at τ=15, producing softened probability distributions that convey richer inter-class relationships. These settings optimize stability and robustness through the knowledge distillation process.

4.6 Baseline Models

We evaluate CTGF-IDS against a range of established benchmark models.Both traditional machine learning and deep learning architectures are included.These baselines enable comprehensive performance comparison across methodologies and complexities.

CL-SKD [34]: A two-stage deep learning framework combining self-supervised learning and knowledgedistillation.It is specifically designed to build lightweight models for IoT environments. This approach ensures strong accuracy while maintaining computational efficiency.

KD-TCNN [13]: This method integrates deep metric learning with knowledge distillation.It produces a compact model suitable for cyber-physical intrusion detection systems.KD-TCNN focuses on balancing accuracy and resource utilization effectively.

LNN [30]: A lightweight deep neural model without knowledge distillation components.It uses depthwise separable convolutions to minimize parameter count and complexity.LNN demonstrates efficiency while maintaining high predictive performance on benchmarkdatasets.

IBYOL-IDS [46]: An intrusion detection system using BYOL (Bootstrap Your Own Latent) learning.It employs self-supervised techniques to capture latent representations efficiently.This model eliminates dependency on labeled data for initial feature extraction.

Fine-tuned Linear SVM [47]: A traditional machine learning method leveraging fine-tuned feature selection techniques. It identifies the most discriminative features for intrusion classification tasks. The selected features train a lightweight, interpretable linear SVM classifier.

These baselines represent diverse strategies in modern intrusion detection research.They collectively provide a strong foundation for evaluating CTGF-IDSperformance.

4.7 Performance Metrics

We employ multiple metrics to assess CTGF-IDS performance objectively.These indicators evaluate accuracy, efficiency, and generalization across intrusion datasets.The adopted metrics are widely accepted in intrusion detection research.

Accuracy (Acc):Measures the ratio of correctly classified samples to total instances.

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (Prec): Quantifies correctly identified positive cases among all predicted positives.

$$\text{Prec} = \frac{TP}{TP + FP}$$

Recall (rec) : It indicates the model effectiveness to discern actual threats

$$\text{rec} = \frac{TP}{TP + FN}$$

F1-Score (F1): Represents the harmonic mean of precision and recall.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Floating Point Operations (FLOPs): Measures computational efficiency of the model architecture. For a 1D CNN, FLOPs are given as:

$$\text{FLOPs} = 2 \times (O \times L) \times K \times C_{\text{in}} \times C_{\text{out}}$$

where OL is the output map dimension, K the kernel size, C_{in} and C_{out} are input and output channels.For dense layers, the FLOPs are computed as:

$$\text{FLOPs} = 2 \times I \times O$$

where I and O represent input and output neurons.These standardized metrics capture architectural efficiency independent of hardware variations [48].They offer consistent benchmarks for evaluating model performance and computational trade-offs.

5. Results and Discussion

This section presents the experimental findings of the proposed CTGF-IDSframework.We evaluate its performance against baseline models and analyze its generalization ability.The focus is on model efficiency, accuracy, and adaptability across multiple datasets.We also assess how knowledge distillation enhances the model's robustness to unseen domains.

5.1Analysis of CTGF-IDS Lightweight Architecture

We begin by analyzing the lightweight characteristics of the proposed architecture.Performance metrics include model size, parameter count, FLOPs, accuracy, precision, recall, and F1.The

objective is to demonstrate high accuracy with reduced computational complexity.We compare CTGF-IDSwith cutting-edge lightweight intrusion detection models.Experiments are conducted for both binary and multiclass classification tasks.

5.1.1 Binary Classification

The binary classification experiment evaluates anomaly versus benign traffic detection performance. Results are summarized in Tables 7 and 8 for multiple datasets .On the BoT-IoT dataset, CTGF-IDS attains an accuracy of 99.8%.This performance highlights its high reliability in identifying abnormal network behaviors. The accuracy achieved equals or surpasses that of strong baselines like CL-SKD.While CL-SKD achieves similar accuracy, it requires higher computational resources. The proposed model exhibits superior efficiency with fewer parameters and lower FLOPs.This confirms CTGF-IDS's suitability for deployment in constrained IoT environments.

All models deliver competitive accuracy values near 99%, indicating robust dataset coverage. However, CTGF-IDS maintain more consistent performance across multiple evaluation metrics. It achieves stable F1 scores and balanced detection across all traffic categories. This consistency demonstrates effective detection capability with minimal false positives.

Table 7: CTGF-IDS vs. Others on BoT-IoT dataset for binary classification

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Model Params	Model Size (KB)
CL-SKD-L [34]	99.00	99.00	99.00	99.00	15,000	56.62
CL-SKD-F [34]	99.00	99.00	99.00	99.00	15,000	56.62
LNN [30]	99.7	–	–	98.81	4,277	181
LSVM Correlation Coefficient Method [47]	96.03	95.83	96.03	95.83	–	–
CTGF-IDS**(Proposed)**	**99.8**	**99.3**	**98.8**	**99.6**	**4,448**	**38**

On the CIC-IDS2017 dataset, CTGF-IDS achieves 99.58% accuracy. It outperforms several strong baselines except the CL-SKD-F variant.The CL-SKD-F achieves a slightly higher accuracy of 99.84%.However, CTGF-IDS achieves this with significantly reduced computational overhead. It requires 70.3% fewer parameters than CL-SKD models on average. Moreover, CTGF-IDS uses around 200 fewer parameters than KD-TCNN student models.This confirms its ability to balance detection precision and computational efficiency.The reduction in model parameters illustrates architectural innovation efficiency.These innovations include partial convolution and channel shuffle-based optimization.The combination enables faster inference without accuracy degradation.Hence, CTGF-IDSprovides real-time intrusion detection for low-power IoT systems.

Table 8: CTGF-IDS vs. Others on CIC-IDS2017 dataset for binary classification

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Model Parameters	Model Size (KB)
CL-SKD-L [34]	99.33	97.38	99.33	97.34	15,000	56.2

CL-SKD-F [34]	99.84	99.81	99.8	99.8	15,000	56.2
IBYOL-IDS [46]	96.7	95.24	95.76	95.5	–	–
KD-TCNN (Student) [13]	97.97	98.02	97.99	97.61	4624	18.1
KD-TCNN (Teacher) [13]	**99.52**	**99.56**	**99.54**	**99.51**	**33,589**	**131.2**
CTGF-IDS**(Proposed)**	**99.58**	**99.38**	**99.28**	**99.56**	**4459**	**34**

5.1.2 Multiclass Classification

In this evaluation, each attack category is treated as a distinct class. We assess performance using BoT-IoT and CIC-IDS2017 datasets under identical configurations.Results highlight model versatility in recognizing multiple attack behaviours simultaneously.Tables 9 and 10 summarize the comprehensive multiclass classification outcomes.

On BoT-IoT, CTGF-IDS demonstrates strong accuracy with minimal computational demand. CL-SKD-L and CL-SKD-F models require nearly 6000× more FLOPs. In contrast, LNN consumes only 1.2× more FLOPs than CTGF-IDS. Regarding parameters, CL-SKD-L and CL-SKD-F use roughly three times more. Meanwhile, LNN employs 144 fewer parameters but yields lower accuracy. CTGF-IDS improves upon LNN by 3.07% and CL-SKD-L as well as CL-SKD-F performs marginally better by 0.78% but at higher computational cost.

The results emphasize an essential trade-off between efficiency and accuracy. Although CL-SKD-F achieves slightly higher accuracy, its complexity limits real-time applicability. CTGF-IDS maintains near-equal accuracy with dramatically reduced computation and storage. This efficiency underscores its suitability for scalable IoT intrusion detection systems.

Table 9:CTGF-IDS vs. Others on BoT-IoT dataset for multiclass classification

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Parameters	FLOPs	Model Size (KB)
CL-SKD-L [34]	99.99	99.00	99.00	99.00	15,000	145297408	56.62
CL-SKD-F [34]	99.99	99.00	99.00	99.00	15,000	145297408	56.62
LNN [30]	96.14	–	96.65	96.65	4,277	23966	181
CTGF-IDS(Proposed)	99.21	99.10	99.05	99.07	4425	21246	38

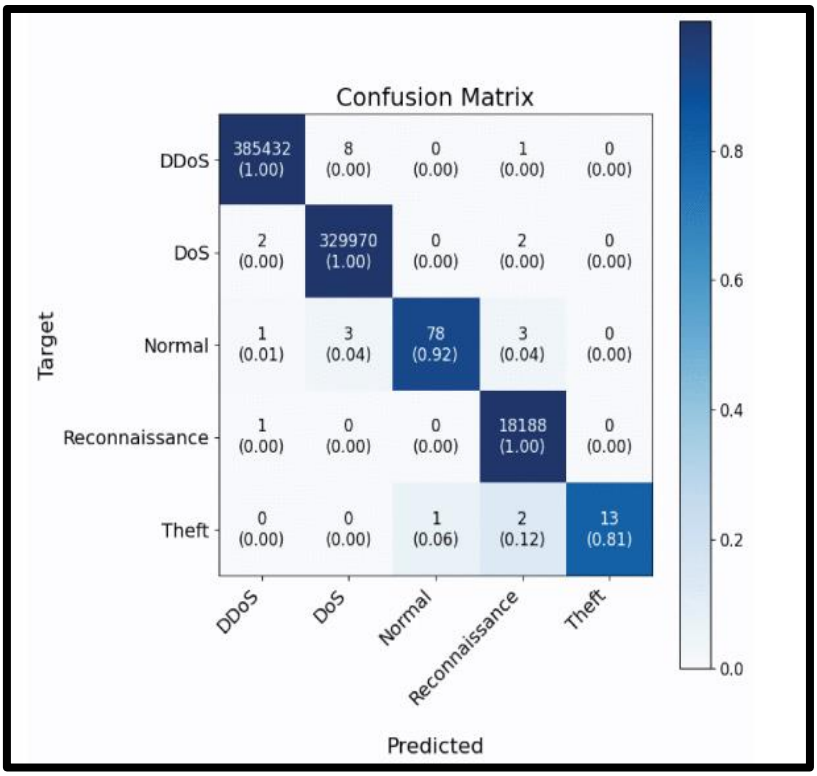

Figure 6: Confusion matrix for multiclass classification on BoT-IoT dataset.

When evaluated on the CIC-IDS2017 dataset, performance consistency remains strong .CTGF-IDS attains 99.58% accuracy, outperforming KD-TCNN and CL-SKD-L. Its accuracy trails CL-SKD-F by only 0.31%, a minimal difference. However, the computational benefits remain substantial for CTGF-IDS .It uses over 6000× fewer FLOPs and 70.3% fewer parameters overall. This highlights its ability to maintain high precision at low computational cost.

The comprehensive metrics in Tables 9 and 10 validate this balance. CTGF-IDS achieves high precision, recall, and F1 across all evaluated classes. This balance demonstrates the model's ability to minimize classification bias. Figures 6 and 7 visualize confusion matrices for both datasets. Diagonal entries remain high, confirming effective identification of attack and benign samples. Off-diagonal elements are sparse, indicating minimal misclassifications or overlaps. Such patterns verify both the reliability and precision of CTGF-IDS classifications.

Table: 10 CTGF-IDS vs. others on CIC-IDS2017 dataset for multiclass classification

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Parameters	FLOPs	Model Size (KB)
CL-SKD-L [34]	99.33	97.38	99.33	97.34	15,000	145297408	56.62
CL-SKD-F [34]	99.80	99.81	99.80	99.80	15,000	145297408	56.62
KD-TCNN [13]	99.52	99.56	99.54	99.51	4,624	–	18.1
CTGF-IDS(Proposed)	99.58	99.54	99.58	99.58	4459	204599	36

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

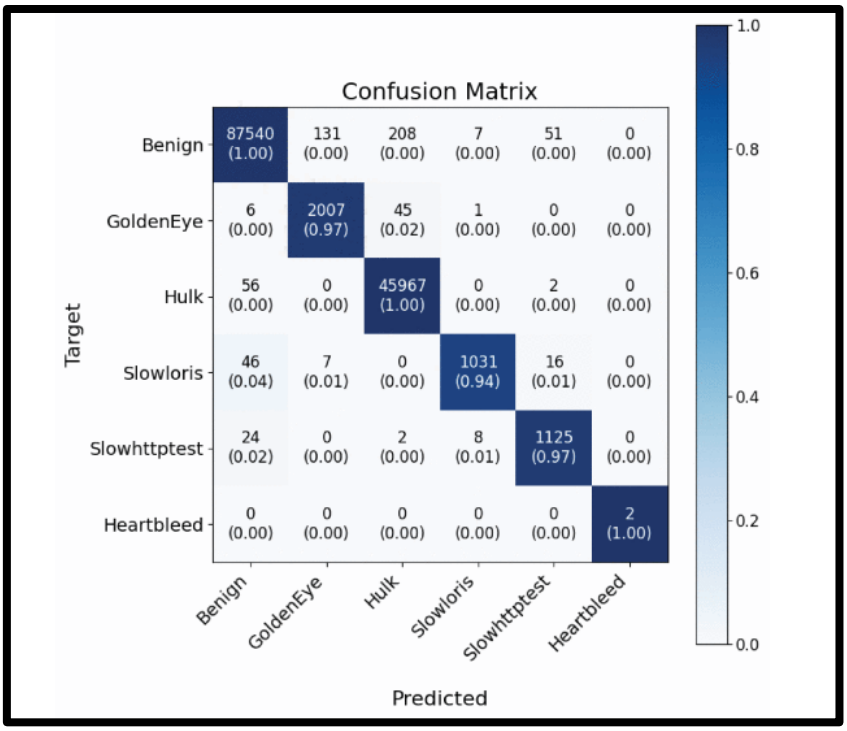

Figure 7: Confusion Matrix for multiclass classification on CIC-IDS2017 dataset dataset.

6. Conclusion

This study introduces **CTGF-IDS**, a lightweight yet highly expressive intrusion detection framework designed for modern IoT environments. By integrating local convolutional feature extraction, global Transformer-based dependency modelling, and an adaptive gated fusion mechanism, CTGF-IDS effectively captures both fine-grained and long-range intrusion patterns. The hybrid architecture is computationally efficient, ensuring real-time operability and low energy consumption—key requirements for deployment on constrained IoT devices.

In combination with a multi-teacher knowledge distillation strategy, CTGF-IDS achieves strong generalization across heterogeneous datasets and diverse network conditions. Distillation enhances the robustness of the student model by transferring soft-label information and intermediate-layer representations from multiple teacher models. This enables CTGF-IDS to maintain stable detection performance under traffic variability and distributional shifts—scenarios commonly encountered in real-world IoT networks.

Experimental evaluations demonstrate that CTGF-IDS consistently deliver competitive or superior accuracy compared with leading state-of-the-art intrusion detection systems, while preserving a compact architectural footprint. Its efficient design minimizes memory usage, reduces inference latency, and lowers computational complexity without sacrificing classification precision. These characteristics position CTGF-IDS as a practical, scalable, and resource-aware solution for large-scale IoT deployments.

Future work will explore adaptive and self-evolving learning mechanisms to improve resilience against continuously emerging threats. Additional research will focus on optimizing CTGF-IDS for distributed and federated IoT settings, enabling secure deployment across multi-

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

layered, heterogeneous networks. Overall, CTGF-IDS represents a significant advancement in lightweight intrusion detection, offering a balanced combination of accuracy, robustness, and operational efficiency for next-generation IoT security infrastructures.

References

[1] S. A. Abdulkareem, C. H. Foh, M. Shojafar, F. Carrez, and K. Moessner, "Network intrusion detection: An IoT and non-IoT-related survey," *IEEE Access*, vol. 12, pp. 147167–147191, 2024.
[2] A. Si-Ahmed, M. A. Al-Garadi, and N. Boustia, "Survey of machine learning-based intrusion detection methods for Internet of Medical Things," *Appl. Soft Comput.*, vol. 140, Jun. 2023, Art. no. 110227.
[3] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to Internet of Things deployment: Survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020.
[4] E.-U.-H. Qazi, T. Zia, M. H. Faheem, K. Shahzad, M. Imran, and Z. Ahmed, "Zero-touch network security (ZTNS): A network intrusion detection system based on deep learning," *IEEE Access*, vol. 12, pp. 141625–141638, 2024.
[5] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019.
[6] F. Mohammadi and M. Saif, "An intrusion detection and mitigation framework for automatic generation control systems," *IEEE Trans. Ind. Cyber-Phys. Syst.*, vol. 2, pp. 412–421, 2024, doi: 10.1109/TICPS.2024.3452681.
[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
[8] A. E. Omolara*et al.*, "The Internet of Things security: A survey encompassing unexplored areas and new insights," *Comput. Secur.*, vol. 112, Oct. 2021, Art. no. 102494.
[9] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun.Technol.*, vol. 32, no. 1, Oct. 2020, Art.no. e4150.
[10] S. T. Slevi and P. Visalakshi, "A survey on deep learning-based intrusion detection systems on Internet of Things," in *Proc. 5th Int. Conf. ISMAC (IoT–Social, Mobile, Analytics, Cloud) (I-SMAC)*, Nov. 2021, pp. 1488–1496.
[11] G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal feature extraction," *J. Netw.Comput.Appl.*, vol. 173, Jan. 2021, Art.no. 102890.
[12] O. Belarbi, A. Khan, P. Carnelli, and T. Spyridopoulos, "An intrusion detection system based on deep belief networks," in *Proc. Sci. Cyber Secur. (SciSec)*, vol. 13580, C. Su, K. Sakurai, and F. Liu, Eds. Cham, Switzerland: Springer, 2022, pp. 377–392, doi: 10.1007/978-3-031-17551-0_25.
[13] Z. Wang, Z. Li, D. He, and S. Chan, "A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art.no. 117671.
[14] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," *arXiv preprint* arXiv:1412.6115, 2014.
[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding," *arXiv preprint* arXiv:1510.00149, 2015.
[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint* arXiv:1503.02531, 2015.
[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
[18] J. Chen *et al.*, "Run, don't walk: Chasing higher FLOPS for faster neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12021–12031.
[19] M. Wang, N. Yang, D. H. Gunasinghe, and N. Weng, "On the robustness of ML-based network intrusion detection systems: An adversarial and distribution shift perspective," *Computers*, vol. 12, no. 10, p. 209, Oct. 2023.

[20] A. H. Halbouni*et al.*, "CNN-IDS: Convolutional neural network for network intrusion detection system," in *Proc. 8th Int. Conf. Wireless Telematics (ICWT)*, Jul. 2022, pp. 1–4.
[21] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art.no. 115524.
[22] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," *Knowl.-Based Syst.*, vol. 279, Nov. 2023, Art.no. 110941.
[23] A. Telikani, J. Shen, J. Yang, and P. Wang, "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 23260–23271, Nov. 2022.
[24] U. C. Akuthota and L. Bhargava, "Transformer-based intrusion detection for IoT networks," *IEEE Internet Things J.*, vol. 12, no. 5, pp. 6062–6067, Jan. 2025.
[25] G.-Q. Zeng*et al.*, "Evolutionary adversarial autoencoder for unsupervised anomaly detection of industrial Internet of Things," *IEEE Trans. Rel.*, early access, Jan. 25, 2025, doi: 10.1109/TR.2025.3528256.
[26] R. V. Mendonça*et al.*, "A lightweight intelligent intrusion detection system for industrial Internet of Things using deep learning algorithms," *Expert Syst.*, vol. 39, no. 5, p. 12917, Jun. 2022.
[27] J.-S. Pan, F. Fan, S.-C.Chu, H.-Q.Zhao, and G.-Y. Liu, "A lightweight intelligent intrusion detection model for wireless sensor networks," *Secur.Commun.Netw.*, vol. 2021, pp. 1–15, Apr. 2021.
[28] R. Zhao, Z. Li, Z. Xue, T. Ohtsuki, and G. Gui, "A novel approach based on lightweight deep neural network for network intrusion detection," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.
[29] S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks," *Expert Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119330.
[30] R. Zhao *et al.*, "A novel intrusion detection method based on lightweight neural network for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9960–9972, Jun. 2022.
[31] S. I. Popoola*et al.*, "Hybrid deep learning for botnet attack detection in the Internet-of-Things networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021.
[32] M. He *et al.*, "A lightweight and efficient IoT intrusion detection method based on feature grouping," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2935–2949, Jan. 2023, doi: 10.1109/JIOT.2023.3294259.
[33] K.-D. Lu *et al.*, "Multi-objective discrete extremal optimization of variable-length blocks-based CNN by joint NAS and HPO for intrusion detection in IIoT," *IEEE Trans. Dependable Secure Comput.*, pp. 1–18, 2025, doi: 10.1109/TDSC.2025.3545363.
[34] Z. Li and W. Yao, "A two-stage lightweight approach for intrusion detection in Internet of Things," *Expert Syst. Appl.*, vol. 257, Dec. 2024, Art. no. 124965.
[35] S. Zhu, X. Xu, J. Zhao, and F. Xiao, "LKD-STNN: A lightweight malicious traffic detection method for Internet of Things based on knowledge distillation," *IEEE Internet Things J.*, vol. 11, no. 4, pp. 6438–6453, Feb. 2024.
[36] S. Yang, X. Zheng, Z. Xu, and X. Wang, "A lightweight approach for network intrusion detection based on self-knowledge distillation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2023, pp. 3000–3005.
[37] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[38] L. Chen, Y. Zhang, Y. Song, Z. Shen, and L. Liu, "LFME: A simple framework for learning from multiple experts in domain generalization," *arXiv preprint* arXiv:2410.17020, 2024.
[39] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1285–1294.
[40] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 357–370, Feb. 2022.
[41] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[42] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, Sep. 2021, Art.no. 102994.
[43] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.
[44] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSp*, vol. 1, Jan. 2018, pp. 108–116.
[45] T. M. Booij*et al.*, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 485–496, Jan. 2022.
[46] Z. Wang, Z. Li, J. Wang, and D. Li, "Network intrusion detection model based on improved BYOL self-supervised learning," *Secur. Commun.Netw.*, vol. 2021, pp. 1–23, Oct. 2021.
[47] J. Azimjonov and T. Kim, "Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors," *Comput.Secur.*, vol. 137, Feb. 2024, Art. no. 103598.
[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.