EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Adaptive Spatio-Temporal Deep Learning for Robust Cyber Threat Detection Across IoT Environments

Narendra Kumar[1,*], Kalai Vani YS[2], Muneshwara M S[3], Chittibabulu Sape[4], V. Subba Reddy[5], Idimadakala Madhavilatha[6]

[1]Department of Computer Science & Engineering, Amity University Jharkhand
[2]Department of Information Science and Engineering, BMS Institute of Technology and Management
[3]Department of Computer Science and Engineering., BMS Institute of Technology and Management Yelahankna, Bengaluru, Karnataka
[4]Department of Computer Science and Engineering, Aditya University, Surampalem
[5]Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation
[6]Department of computer science and engineering, SV College of Engineering, Karambadi road, Tirupati

Abstract

In interconnected systems such as the Internet of Things (IoT), industrial control systems, and smart cities, real-time intrusion detection is crucial in smart network environments. It analyses network traffic in real time, enabling rapid detection and mitigation of cyber threats. With the help of artificial intelligence, including deep learning, real-time intrusion detection systems (IDS) can spot suspicious patterns, adapt to new-fangled attack vectors, and keep latency low, all without sacrificing system performance or reliability. This paper addresses intrusion detection in smart network environments by introducing Net Sentry DL, a deep learning framework. To extract spatio-temporal features and improve interpretability, the model uses a combination of CNNs, TCNs, and attention-guided fusion. Data imbalance and noise are addressed using an entropy-based pre-processing method and a class-preserving algorithm called Synthetic Minority Oversampling Technique (SMOTE). The UNSW-NB15, BoT-IoT, and TON_IoT benchmark datasets are used to assess Net Sentry DL. It surpasses models such as SVM, Random Forest, LSTM, GRU, and CNN in binary classification, reaching up to 0.99 accuracy and 0.98 F1-score on BoT-IoT. With TON_IoT, it achieves an accuracy of 0.95, and with BoT-IoT, up to 0.97, in multi-class configurations. Using SHAP, attention heatmaps, and gated fusion visualisations, the model exhibits strong explainability and robust generalisation, as demonstrated by cross-dataset testing. Through ONNX conversion and low quantisation loss (0.7%), it efficiently deploys and achieves low inference time (37ms/sample). The significance of each module, particularly CP-SMOTE and the TCN-attention combination, has been confirmed by ablation studies. For ever-changing IoT-based infrastructures, Net Sentry DL demonstrates competitive accuracy, interpretability, and deployment efficiency for intrusion detection.

Keywords: Real-time Intrusion detection; Convolutional Neural Networks; Temporal Convolutional Networks; Synthetic Minority Oversampling Technique; Industrial control systems; Internet of Things.

Received on 04 November 2025, accepted on 16 May 2026, published on 29 June 2026

doi: 10.4108/10791

*Corresponding author. Email: nkumaracademia@gmail.com

1. Introduction

One of the most important aspects of digital transformation in this age of hyper-connectivity is smart networks, which include IoT devices, industrial sensors, and autonomous communication services [1]. Cybersecurity has never faced such a surge in threats, especially in real-time intrusion detection [2].

Modern cyber threats are too large, diverse, and dynamic for traditional rule-based systems to handle. Intelligent, adaptive, and interpretable intrusion detection systems (IDS) are crucial due to the increasing complexity of adversarial attacks, which encompass a wide range of tactics, from botnet propagation to advanced zero-day vulnerabilities [3]. Two new paradigms that show promise for solving this issue are learning. Static attack patterns can be effectively detected by ML-based intrusion detection systems such as XGBoost, Random Forests (RF), and Support Vector Machines (SVM) [4]. When asked to capture dependencies within network traffic that occur sequentially or over time, these models frequently fail [5]. This is something that deep learning architectures such as GRU and LSTM networks try to tackle by learning temporal patterns [6]. Despite this, they have low parallelism, high training costs, and vanishing gradients [7]. A lot of recent work has focused on developing hybrid models that combine spatial and temporal learning capabilities to circumvent these limitations [8]. Byte size, flags, and port activity are packet-level characteristics that Convolutional Neural Networks (CNNs) successfully captured spatial correlations among [9]. One alternative to RNNs for temporal modelling is the Temporal Convolutional Network (TCN), which offers dilated convolutions, large receptive fields, and better parallelism than RNNs [10]. It is possible to extract detailed representations of spatio-temporal intrusion signatures by combining CNN and TCN modules. However, contextual feature prioritisation is still severely underdeveloped [11]. While some feature sets or time steps may be more important than others for differentiating between safe and harmful activity, many IDS models treat all features equally [12].

By enabling the model to dynamically assign weights to features based on relevance, transformer-inspired self-attention mechanisms provide a strong solution [13–14]. Furthermore, these signals can be refined using gated fusion layers, which improve the model's accuracy and interpretability [15]. Although recent intrusion detection studies have employed hybrid deep learning architectures such as CNN-TCN-Attention networks, most existing models primarily emphasise classification accuracy while overlooking several practical limitations of real-world IoT security environments. Many frameworks do not effectively address severe class imbalance, preserve minority attack semantics during oversampling, or adaptively balance spatial and temporal threat cues under dynamic traffic conditions. In addition, lightweight deployment for edge devices and transparent decision interpretability remain insufficiently explored. To overcome these gaps, the proposed Net Sentry DL framework integrates class-preserving CP-SMOTE, dynamic threat-aware fusion gating, confidence-guided attention learning, and deployment-efficient optimisation for robust real-time intrusion detection.

The Net Sentry DL architecture incorporates a CNN-TCN backbone and an attention-guided fusion module to meet these requirements. This multi-layered design enables strong learning from both immediate and distant dependencies, with adaptive emphasis on signals relevant to the current context. New pre-processing innovations introduced by Net Sentry DL—entropy-based quantile encoding and Class-Preserving Synthetic Minority Oversampling Technique (CP-SMOTE) — help reduce dataset issues such as class imbalance and feature redundancy. By following these steps, you can ensure that underrepresented attack classes are better represented and that variability in network behaviour is preserved.

The selection of evaluation datasets—UNSW-NB15, BoT-IoT, besides TON_IoT—demonstrates a dedication to comparing results across diverse and realistic traffic scenarios. These datasets contain a diverse array of attack vectors recorded under different network conditions. These include DDoS, reconnaissance, backdoors, and data exfiltration, among others. In both the binary and multi-class scenarios, experimental results reveal that Net Sentry DL outperformed selected baseline models. On BoT-IoT (binary), Net Sentry DL attains an accuracy of 0.99, besides an F1-score of 0.98; on TON_IoT (multi-class), it scores an accuracy of 0.95 and an F1-score of 0.94. These metrics show that, compared to traditional models, there is a consistent improvement of 3-10%.

A main goal of Net Sentry DL is to ensure efficient deployment, alongside raw performance. The model was converted to the ONNX format, making it suitable for use in embedded systems and edge environments. With relatively low utilisation of GPU (68%) and CPU (42%), it attains an inference speed of 37 milliseconds per sample. With a mere 0.7% decrease in accuracy following ONNX conversion, the model also shows remarkable quantisation stability. Because of these features, it is ideal for smart grid infrastructure and industrial control systems, both of which are very sensitive to latency. One more thing Net Sentry DL has going for it is explainability. Incorporating features such as attention heatmaps, SHAP value analysis, and LIME-based explanations helps cybersecurity experts understand model predictions and verify their reliability. Error heatmaps and gated fusion timelines are visualisation tools that help us understand how the model behaves across different network states and phases of an attack. These features are useful for forensic analysis after an incident and for operational decision-making.

In addition, experiments conducted on different datasets confirm the framework's ability to generalise. To illustrate resilience to domain shifts and unseen traffic patterns, training on UNSW-NB15 and testing on BoT-IoT, for example, results in a performance drop of only 4-5%. Because of this, Net Sentry DL is great for deployments in varied or ever-changing environments where labelled data isn't always accessible. Lastly, ablation studies validate the importance of every part of the building. Eliminating

attention-guided fusion causes even steeper drops in F1-score, while removing TCN causes a 3-4% drop. The model's ability to learn from minority classes is significantly affected by the elimination of CP-SMOTE, indicating that CP-SMOTE improves detection sensitivity.
Below is the outline for the remainder of the paper: Section 2 reviews the relevant literature; Section 3 outlines the methodology; Section 4 discusses the analysis of the results; Section 5 concludes.

1.1. Major Contributions of This Work

- A novel Net Sentry DL framework is proposed for real-time intrusion detection in IoT environments by integrating CNN, TCN, and attention-guided fusion mechanisms to simultaneously capture spatial and temporal attack characteristics.
- A Class-Preserving Synthetic Minority Oversampling Technique (CP-SMOTE) is introduced to alleviate class imbalance while preserving minority attack semantics and temporal consistency.
- An attention-guided dynamic threat-aware fusion gate is developed to selectively emphasize critical temporal-spatial features, thereby improving detection performance and model interpretability.
- The proposed framework incorporates explainable AI mechanisms including SHAP analysis, attention heatmaps, gated fusion visualization, and error heatmap analysis to improve transparency of intrusion detection decisions.
- Lightweight deployment optimization through pruning, quantization, and ONNX conversion is incorporated to support real-time deployment on resource-constrained IoT and edge devices.
- Extensive experiments on UNSW-NB15, BoT-IoT, and TON_IoT datasets demonstrate superior binary and multi-class intrusion detection performance compared with conventional machine learning and deep learning baselines.
- Cross-dataset validation, ablation studies, statistical significance testing, and deployment analysis verify the robustness, generalization capability, and practical applicability of the proposed Net Sentry DL framework.

2. Related Works

Using two machine learning algorithms—Random Forest and Support Vector Machine—Luqman, M., et al., [16] suggest an intelligent intrusion detection system for IoT networks. By balancing 11 classes in the BoT-IoT dataset, it has achieved high efficiency and reduced overfitting. Handling missing values, one-hot encoding, and feature scaling for binary and multi-class classification were among the preprocessing techniques used to prepare the dataset for analysis. Using two hyperparameter tuning methods, RF and SVM models underwent a crucial feature extraction and selection process. The proposed model efficiently trains SVM and RF using three cross-validation techniques with 10-fold cross-validation. A binary classification accuracy of 99.60% and a multi-class classification accuracy of 98.31% were accomplished through machine learning. User Datagram Protocol and Transmission Control Protocol packets constitute the anomalous traffic in the UNSW BoT-IoT dataset used by deep learning algorithms. Using feature engineering based on common features besides the deep learning model LSTM for training, to merge two important network datasets, UNSW BoT-IoT and NB-15, to address this imbalance. To validate the model with 10 iterations, we achieved a 99.89% accuracy for multi-class classification and 99.97% accuracy for single-class classification after feature extraction.
Combining CNNs for spatial feature extraction with four RNN variants—Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (BiGRU)—to capture temporal features is the IDS architecture proposed by Jablaoui, R., and Liouane, N. [17]. These networks work in tandem to detect and classify malicious cyberattacks in IoT network traffic more accurately than previous approaches. To test our model on the NF-UQ-NIDS, a large-scale Netflow dataset for binary classification that combines information from several sources (NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018). To use several metrics, including recall, accuracy, precision, F1 score, FAR, and AUC, to assess the presentation of the models to suggest. Four different models were tested in various datasets as part of a thorough comparative study. The study highlights the potential of integrating spatial, as well as temporal, DL models for more sophisticated network security applications, showing that this approach can significantly enhance detection accuracy while reducing false alarm rates. Hence, the remarkable results demonstrate the efficacy of combining CNNs with various RNN variants for IDS in IoT networks, providing a robust defence for IoT ecosystems against security risks.
A framework for intrusion detection in IoT, based on Feature Selection alongside Large Language Models (LLMs), is proposed by Ma, H., et al. [18]. It all comes down to a multi-stage feature selection algorithm that combines mRMR and a PCC-improved Covariance Matrix Adaptation Strategy algorithm for optimal feature extraction. To reduce redundancy more effectively, this algorithm considers mutual information and collinearity among features and uses the CMA-ES algorithm for feature search. To then use these representative features to improve LLMs and make more attack samples. The computational cost of fine-tuning is effectively reduced, and higher-quality samples are produced using this approach. Also, to boost detection accuracy, use the FL-function-improved LightGBM as the classifier. To test our system on five Internet of Things (IoT) intrusion detection datasets: CIC-ToN-IoT, NF-CSE-CIC-IDS2018-v2, NF-BoT-IoT-v2, NF-UNSW-NB15-v2, and NF-CSE-CIC-IDS2018-04. The experimental results show that FSLLM reduces redundant features by more than 80% while

achieving the same level of accuracy as or even better than state-of-the-art approaches.

In their groundbreaking work, Kamal and Mashaly [19] introduce an intrusion detection method designed for Internet of Things (IoT) networks—one based on anomalies. The suggested model uses a state-of-the-art hybrid architecture that seamlessly merges a CNN and an MLP, enabling accurate detection and categorisation of binary and multi-class IoT network traffic. For the MLP to perform effective classification, the CNN component must first extract and refine features from network traffic data. For binary classification, the model uses ADASYN-SMOTE, and for multiclass classification, it uses advanced ADASYN. Additionally, it uses edited nearest neighbours (ENN) in conjunction with class weights to address class imbalance further. The CNN-MLP architecture has been painstakingly designed to reduce false positives, improve real-time threat detection, and accurately identify previously unseen cyber intrusions. Using the IoT-23 and NF-BoT-IoT-v2 datasets, the model's efficacy was tested extensively. The model achieved a two-stage binary classification accuracy of 99.94% on the IoT-23 dataset, a multiclass classification accuracy of 99.99% when the normal class was excluded, and a single-phase multiclass classification accuracy of 99.91% when the normal class was included. In the dual-phase binary classification paradigm, the model achieved an exceptional 99.96% accuracy for the normal class; in single-phase multiclass classification, including the normal class, the model achieved 98.02% accuracy. In both binary and multiclass classification, our model consistently achieves high levels of accuracy, precision, recall, and F1 score. This establishes it as a strong solution for securing IoT networks.

Leni, A. E. S., et al. [20] introduced a refined deep learning model to enhance cyber-attack detection in IoT networks. With little collateral damage, this model accurately categorises various forms of cyberattacks. The feature selection process involves selecting the most relevant subset of the original network traffic features using a wrapper-based dwarf mongoose optimisation algorithm (W-DMO). The features are then classified, and various types of attacks are labelled using a hybrid triple-attention deep neural network-assisted BiLSTM model (TDeepBiL). The suggested method is evaluated using multiple performance metrics, including F1-score, recall, accuracy, and precision. Compared with existing models, the proposed model achieves a very high accuracy of 99.44% on the UNSW-NB 15 dataset and 98.6% on the ToN-IoT dataset. When it comes to detecting cyberattacks, the presented model is much better.

To carry out the best possible feature extraction and fusion procedures, Silivery, A. K., et al. [21] introduced a Dual Path Feature Extraction Network. The Neural Architecture Search Network is subsequently used to detect and classify attacks using the combined features. Also, the Conditional Network fixes the data imbalance problem. Three datasets, BoT-IoT, ToN-IoT, and IoT-23, are used to measure the effectiveness of the proposed framework. Results from the experiments demonstrate that the proposed system outperforms its rivals and performs well in IoT settings.

A new deep learning framework, DC-NFC, is introduced by Rehman, A. et al. [22] to improve the privacy and security of NFC communications in Internet of Things (IoT) settings. The AMOL enforces end-to-end privacy constraints, the CE captures complex temporal and spatial patterns, and the ATF module detects threats in real time and dynamically adapts models; these three innovative components are integrated into the proposed framework. The four benchmark datasets used in the experiments were UNSW-NB15, Bot-IoT, TON-IoT Telemetry, and Edge-IIoTset. Highlighting the framework's resilience in improving NFC security besides privacy in diverse IoT environments, the consequences show substantial improvements in security metrics across all datasets. With a latency of only 20.53 s for 1000 devices on the UNSW-NB15 dataset, the simulation results demonstrate the framework's ability to process data in real-time.

A Zero-Day attack detection NIDS based on Deep Reinforcement Learning (DRL) is proposed by Alam, K. et al. [23]. To improve the DRL agent's learning capabilities, employ a stacked LSTM architecture. Since there is a dearth of zero-day attack datasets, we employ multiple oversampling strategies to address class imbalance. To employ a number of the most popular NIDS benchmark datasets that collectively address numerous attack types, including reconnaissance, distributed denial of service, injection, brute force, DOS, backdoor, benign traffic, and infiltration. As an illustration, assign a value of 1 to attacks and a value of 0 to benign traffic. Then, to separate the training dataset from the test dataset, exclude specific types of attacks, such as DoS and Backdoor. Because they are completely hidden during training, these attack types are called zero-day attacks. To further evaluate the efficacy of various data balancing techniques on our DRL agent's performance and compare K-means SMOTE, SMOTE, Borderline-SMOTE, and ADASYN. Afterwards, we aim to prove our agent's strength by successfully zeroing in on known and unknown attacks across multiple datasets. Table 1 presents the comparative analysis of existing techniques.

Table 1. Comparative Analysis of Existing IDS Methods

Ref	Method	Dataset	Strength	Limitation
[16]	RF + SVM	BoT-IoT	High classification accuracy	Limited temporal dependency learning
[17]	CNN + RNN	NF-UQ-NIDS	Spatial-temporal modeling	Higher computational complexity
[18]	FSLLM	Multiple IoT datasets	Strong feature reduction	Limited deployment discussion
[19]	CNN-MLP	IoT-23, NF-BoT-IoT	High detection accuracy	Explainability not addressed

[20]	W-DMO + TDeepBiL	UNSW-NB15, ToN-IoT	Effective attack detection	Higher model complexity
[21]	Dual Path Network	BoT-IoT, ToN-IoT, IoT-23	Feature fusion capability	Limited interpretability
[22]	DC-NFC	UNSW-NB15, Edge-IIoTset	Real-time detection	Focused on NFC security
[23]	DRL-LSTM	Multiple datasets	Zero-day attack detection	High training overhead
[28]	ML-Based IoT Botnet Detection Framework	IoT Botnet Dataset	High botnet detection accuracy (99.50%) with optimized feature engineering	Primarily focused on botnet attacks; lacks explainability and deployment optimization
[29]	Hybrid RF-BiLSTM Framework	Aposemat IoT-23	Strong temporal traffic modeling and feature relevance learning (99.87%)	Increased computational overhead and limited interpretability support
[30]	CapsNet + Teamwork Optimization Algorithm (TOA)	BoT-IoT / IoMT Environment	Effective detection of known and unknown attacks (98.37%)	Focused mainly on IoMT environments; lacks cross-domain validation and deployment analysis
Proposed	**Net Sentry DL**	**UNSW-NB15, BoT-IoT, TON_IoT**	**CP-SMOTE + CNN-TCN + Attention-Guided Fusion + SHAP Explainability + ONNX Deployment Optimization + Cross-Dataset Generalization**	**Moderately higher training complexity due to multi-stage architecture**

Recent studies have demonstrated significant progress in IoT intrusion detection through machine learning, deep learning, and hybrid frameworks. Methods such as RF-SVM, CNN-RNN, CNN-MLP, DRL-LSTM, and hybrid optimization-based models have achieved high detection accuracy across various benchmark datasets. Furthermore, recent studies published in *EAI Endorsed Transactions on Internet of Things* have reported strong performance using machine-learning-based botnet detection frameworks [28], RF-BiLSTM temporal learning models [29], and CapsNet-based IoMT intrusion detection systems [30]. However, most existing approaches primarily focus on classification performance and often overlook critical requirements such as class-preserving imbalance handling, explainable decision-making, lightweight deployment, adaptive spatio-temporal fusion, and cross-dataset generalization. To address these limitations, the proposed Net Sentry DL framework integrates CP-SMOTE, CNN-TCN feature extraction, attention-guided dynamic fusion, SHAP-based explainability, and ONNX-enabled deployment optimization within a unified architecture, providing a comprehensive solution for real-time intrusion detection across heterogeneous IoT environments. From the comparative analysis, it can be observed that existing intrusion detection frameworks primarily focus on classification accuracy, feature selection, or temporal modeling. However, limited attention has been given to simultaneously addressing class imbalance, explainability, lightweight deployment, and adaptive spatio-temporal threat representation. Furthermore, many studies do not provide deployment-oriented optimization or interpretable decision support. To address these limitations, the proposed Net Sentry DL framework integrates CP-SMOTE, CNN-TCN feature extraction, attention-guided fusion, explainability analysis, and ONNX-based deployment optimization within a unified architecture.

3. Overview of Net Sentry DL Architecture

3.1. Design Principles and Motivation

Accuracy is essential, but real-time IDS in the dynamic world of smart networks, which include sensors, autonomous services, and Internet of Things (IoT) devices, requires flexibility, efficiency, and interpretability. To address these concerns, the Net Sentry DL framework incorporates deep learning modules that complement one another to detect, identify, and categorise intrusion patterns in a low-latency, resource-limited setting. The proposed model's workflow is exposed in Figure 1.

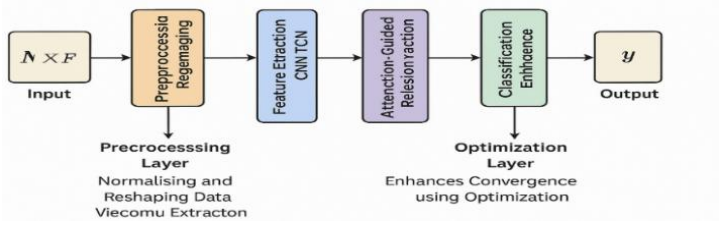

Figure 1. Workflow of the proposed model

Unlike traditional rule-based or shallow learning methods, Net Sentry DL adopts a multi-stage deep learning pipeline. The framework aims to capture the non-linear and temporal relationships between features using Temporal Convolutional Networks (TCNs), while also employing attention-based fusion to ensure relevant information is prioritised. Each component is optimised for edge deployment via model pruning and ONNX conversion.

3.2. Layered Architecture Description

The architecture comprises the following components:

- **Pre-processing Layer**: Normalises, augments, and reshapes data using Class-Preserving SMOTE and entropy quantisation.
- **Feature Extraction Block**: Integrates CNN and TCN to capture spatial-temporal patterns.
- **Attention-Guided Fusion Module**: Learns contextual relevance across features.

- **Classification Block**: Makes final multi-class predictions using Focal Loss-enhanced softmax.
- **Optimisation Layer:** Enhances convergence using gradient-based adaptive optimisers.

Let $X \in R^{N\times T\times F}$ denote the network input data,
where: N: batch size, T: temporal window or sum of time steps and F: sum of features (e.g., protocol flags, bytes, packet rate). To define the goal as learning a mapping function:

$$f_\theta : X \to Y \quad where \ Y \in R^{N\times C} \tag{1}$$

Here: Y: one-hot encoded labels across C classes (e.g., benign, DoS, probe, botnet), θ: trainable parameters of the model.

3.3. Mathematical Pipeline of Net Sentry DL

The architecture unfolds as a sequential mapping of transformations:

$$f_\theta = f_{class} * f_{fusion} * f_{extract} * f_{prep} \tag{2}$$

Where f_{prep} pre-processing and SMOTE transformation, $f_{extract}$ is CNN-TCN feature extraction, f_{fusion} attention-based fusion and f_{prep} final classification block. Each transformation is defined below in a modular and mathematical format.

3.4. Pre-processing Function f_{prep}

When dealing with complex datasets such as UNSW-NB15 [24], BoT-IoT [25], and TON_IoT [26], the text emphasises the critical importance of a robust preprocessing pipeline for real-time intrusion detection systems (IDS). Problems common in these datasets include erratic packet sampling intervals, class imbalance (e.g., a majority of benign samples), and features that are either too noisy or too redundant to detect attacks. Patterns unfold at different temporal granularities due to multiscale temporal behaviours, which further complicates simple modelling.

In response to these issues, the Net Sentry DL framework implements a thorough preprocessing plan to enhance data quality, reduce learning bias, and preserve relevant temporal relationships. Training is stabilised, and convergence is accelerated by first applying normalisation techniques to bring heterogeneous feature scales to a common range. Second, the framework employs Class-Preserving SMOTE (CP-SMOTE), a refined oversampling technique, to address data imbalance. In contrast to generic SMOTE, CP-SMOTE preserves temporal order and class boundaries, ensuring that the synthetic data is realistic and consistent with the semantics of network traffic. Incorporating statistical descriptors reflecting variability and distributional structure into each feature vector, such as Shannon entropy and quantile encoding, is another crucial improvement. Because of this, the model has irregularities in patterns that would otherwise appear normal. The subsequent CNN and TCN-based feature extraction layers rely on these pre-processing steps to minimise computational overhead and overfitting while retaining and amplifying meaningful patterns.

Given an input $x \in R^{T\times F}$The preprocessing transformation includes:

- Quantile-based normalisation,
- Shannon entropy encoding $H(x)$,
- Synthetic oversampling using CP-SMOTE.

Let $x_i \in R^F$ be a single time-step vector. The entropy for each feature f is defined as:

$$H_f(x) = -\sum_{k=1}^{K} p_k \log p_k \,,\ p_k = \frac{freq(x_f = v_k)}{T} \tag{3}$$

p_k: empirical frequency of a discrete value v_k, K: number of bins used for quantisation.

This transformation augments feature vectors. $x_i \to \hat{x}_i \in R^{F+F'}$, where F' denotes appended entropy-based attributes.

The CP-SMOTE oversampling is then applied on minority classes using:

$$\hat{x}_{synthetic} = x_{minor} + \lambda.(x_{nearest} - x_{minor}) \tag{4}$$

$\lambda \sim U(0,1)$: random scalar, $x_{nearest}$: k-nearest neighbour in feature space. This step enhances class balance without compromising real-time adaptability, which is especially important for high-imbalance datasets like BoT-IoT.

3.4.1 Class-Preserving CP-SMOTE Strategy

In highly imbalanced intrusion detection datasets, minority attack categories such as reconnaissance, backdoor, shellcode, or user-to-root traffic often contain significantly fewer samples than benign or dominant attack classes. Direct training on such skewed distributions may bias the classifier toward the majority classes, reducing sensitivity to rare but security-critical threats. To alleviate this limitation, the proposed Net Sentry DL framework employs a Class-Preserving CP-SMOTE strategy that extends conventional oversampling by generating synthetic minority instances while preserving local class structure, temporal consistency, and feature-space continuity. Unlike ordinary SMOTE, which interpolates samples solely based on Euclidean proximity, CP-SMOTE constrains synthetic generation to trusted neighbourhood regions within the same attack class, thereby reducing class overlap and noisy sample creation.

Let the minority class sample set be represented as $\mathcal{S}_m = \{x_1, x_2, \ldots, x_n\}$, where each $x_i \in \mathbb{R}^d$denotes a d-dimensional feature vector after normalisation and entropy augmentation. For each minority instance x_i, a set of k Nearest neighbours from the same class are identified as $\mathcal{N}_k(x_i)$. A neighbour $x_j \in \mathcal{N}_k(x_i)$is then selected for synthetic interpolation. To preserve class consistency, the local density reliability of the neighbourhood is first estimated as

$$\omega_i = \frac{1}{k}\sum_{j=1}^{k} \exp\ -\left(\left\|x_i - x_j\right\|_2\right) \tag{5}$$

where $\omega_i \in (0,1]$denotes the neighbourhood trust coefficient, and larger values indicate compact and homogeneous minority regions. This coefficient ensures that dense and reliable attack clusters receive greater oversampling priority than sparse boundary zones.

The synthetic sample $x_{new}$is generated by weighted interpolation between $x_i$and $x_j$as

$$x_{new} = x_i + \lambda\, \omega_i (x_j - x_i) \quad (6)$$

where $\lambda \sim U(0,1)$is a uniformly distributed random scalar controlling interpolation distance. Because the displacement term is modulated by ω_iThe generated sample remains closer to stable minority manifolds and avoids drifting toward majority decision regions.

To further retain sequential behaviour in traffic records, the temporal descriptor associated with each minority sample is preserved during generation. If $t_i$and $t_j$denote the temporal indices of $x_i$and x_j, the synthetic temporal component is computed as

$$t_{new} = t_i + \lambda(t_j - t_i) \quad (7)$$

thereby maintaining realistic progression patterns for burst traffic, scanning intervals, or session-based attack traces. The final augmented sample is represented as

$$\tilde{x}_{new} = [x_{new}, t_{new}] \quad (8)$$

where [.] denotes the concatenation of feature and temporal descriptors. For each minority class c, the number of synthetic samples required is determined by imbalance severity:

$$G_c = \max(0, N_{max} - N_c) \quad (9)$$

where $N_c$is the current number of samples in class c, and $N_{max}$is the sample count of the largest class. This adaptive formulation balances all classes without unnecessary oversampling of already sufficient categories.

The CP-SMOTE output set is therefore expressed as

$$\mathcal{D}^* = \mathcal{D} \cup \{\tilde{x}_{new}^{(1)}, \tilde{x}_{new}^{(2)}, \ldots, \tilde{x}_{new}^{(G_c)}\} \quad (10)$$

where $\mathcal{D}$is the original training dataset and $\mathcal{D}^*This\ equation\ isThis\ This\ equation\ is$is the balanced training set used for model learning. Through this mechanism, Net Sentry DL improves recall for minority intrusion classes, stabilises decision boundaries, and reduces false negatives without degrading majority-class discrimination. This behaviour is particularly beneficial for datasets such as BoT-IoT and TON_IoT, where attack distributions are highly skewed and conventional learning methods tend to overlook rare threat categories.

3.5. Feature Extraction $f_{extract}$

In the field of real-time intrusion detection, once the input sequence $X_{windowed} \in R^{T \times D}$ is preprocessed, the next essential step is to extract meaningful patterns embedded across both spatial (feature) and temporal (sequence) dimensions. Intrusion signatures often exhibit complex dependencies such as combinations of protocol behaviours (e.g., SYN+ACK+payload size) and irregular timing bursts (e.g., botnet traffic or slow port scans). To effectively capture these multifaceted patterns, Net Sentry DL integrates a hybrid deep learning component that combines networks.

CNNs are adept at identifying local spatial correlations across feature vectors within individual packets or fixed windows, making them suitable for analysing concurrent protocol and flag interactions. Meanwhile, TCNs are specialised for learning long-range temporal dependencies through dilated convolutions, offering superior performance in modelling causally structured sequences compared to recurrent models, without the training instability or latency overhead. The TCN architecture maintains temporal ordering, crucial for real-time applications. This hybrid design offers both efficiency and scalability by avoiding the computational complexity of attention-heavy models while maintaining real-time suitability. Thus, CNN+TCN forms a synergistic module that enables robust intrusion-signature learning in dynamic smart-network environments.

3.5.1 Spatial Feature Encoder: CNN

A 1D CNN is employed over temporal packets to extract spatial correlations among features. The convolution operation for a layer l is defined as:

$$h^{(l)} = \sigma\left(W^{(l)} * h^{(l-1)} + b^{(l)}\right) \quad (11)$$

Where: *: 1D convolution over time, $W^{(l)} \in R^{k \times F_{in} \times F_{out}}$: convolution kernel, $b^{(l)}$: bias vector and $\sigma(\cdot)$: ReLU or ELU activation. The output feature map is denoted.

$$H_{CNN} \in R^{T' \times F''} \quad (12)$$

where: T′: reduced temporal dimension, F″: number of learned filters. CNNs capture inter-feature dependencies across packets efficiently and have a reduced memory footprint compared to full attention maps.

3.5.2 Temporal Dependency Modelling: TCN

To model long-range temporal sequences, we introduce a Temporal Convolutional Network (TCN) block. The dilated causal convolution is:

$$y_t = \sum_{i=0}^{k-1} \omega_i \,.\, x_{t-d.i} \quad (13)$$

Where, d: dilation factor (exponential: $d = 2^l$), k: kernel size, x_t: input at time t and w_i: learned kernel weight. This structure allows exponential growth of the receptive field. The output $H_{TCN} \in R^{T'' \times F''}$ captures time-oriented features like burst anomalies and botnet periodicity. It ensures causal modelling—each output at the period. t be contingent only on t and previous time steps, critical for real-time prediction.

3.5.3 Unified Representation

The outputs from CNN and TCN are concatenated and passed through a projection layer:

$$Z = \phi(W_z[H_{CNN}; H_{TCN}] + b_z) \quad (14)$$

Where, [·;·]: concatenation operator, $W_z \in R^{(F''+F'') \times F_d}$: projection weights, $\phi(\cdot)$: GELU activation. The result $Z \in R^{T'' \times F_d}$ forms the spatio-temporal feature tensor, feeding into the fusion module.

3.6. Dynamic Threat-Aware Fusion Gate f_{fusion}

To prioritise relevant temporal-spatial patterns, to employ a self-attention mechanism inspired by Transformer encoders:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (15)$$

Where: $Q, K, V \in R^{T \times d_k}$: learned query, key, value vectors from Z, d_k: dimension of key vectors. This produces a contextualised tensor. $Z_{att} \in R^{T'' \times F_d}$, emphasising time-steps highly correlated with intrusion patterns.
Additionally, to use a gated fusion mechanism:

$$F = \sigma(W_g Z + b_g) \odot Z_{att} \quad (16)$$

$\odot$: element-wise multiplication (gating), σ: sigmoid activation, F: final fused representation (context-aware and filtered). It assigns dynamic importance to patterns across time and feature maps, enhancing interpretability and reducing false alarms.

3.7. Classification Function f_{class}

After spatial-temporal features are extracted and refined through the CNN-TCN backbone and attention-guided fusion mechanism, the resultant high-level feature vector $F_{final} \in R^{D_z}$ encapsulates the most discriminative temporal-spatial semantics of the network traffic. The classification module's primary objective is to translate this dense representation into class-wise probabilistic predictions, enabling the identification of various intrusion types in a multi-class setting. To ensure robust learning and generalisation, particularly in the presence of highly distributed data, the module is designed to maintain differentiability throughout the learning process, allowing seamless integration into an end-to-end optimisation pipeline.
The transformation begins with a fully connected dense layer, which linearly maps the latent class score vector (logits). These logits are then passed through a softmax function, which normalises the scores into a valid likelihood distribution over all classes, making the output interpretable and suitable for probabilistic decision-making. To mitigate the adverse effects of class imbalance—common in real-world intrusion datasets—focal loss is employed. This loss function dynamically down-weights easy, majority-class examples, emphasising hard or minority-class samples during training. Thus, the classification block is mathematically sound, computationally efficient, and tailored for real-time intrusion detection under class-skewed conditions. The final fused output is globally pooled and passed to a layer with a softmax activation:

$$\hat{y} = softmax(W_f F_{avg} + b_f) \quad (17)$$

Where, $F_{avg} = \frac{1}{T''} \sum_{t=1}^{T''} F_t$, global average over time, $W_f \in R^{F_d \times C}, b_f \in R^C, \hat{y} \in R^C$ predicted class probabilities.

3.8. Loss Function and Training Objective

To employ a weighted Focal Loss to counter class imbalance and overconfidence:

$$\mathcal{L}_{focal} = -\sum_{c=1}^{C} a_c (1 - \hat{y}_c)^{\gamma} y_c \log(\hat{y}_c) \quad (18)$$

Where, a_c is class weight (inverse frequency), γ is focusing parameter (typically 2), y_c is ground truth and $\hat{y}_c$ predicted probability for class c. It reduces loss contribution from easy samples and focuses training on hard misclassified intrusions.

3.9. Lightweight Edge Deployment Optimisation

To enable practical, real-time deployment in IoT gateways, embedded controllers, and resource-constrained smart network devices, the trained Net Sentry DL framework is further optimised through a lightweight edge-deployment stage. After learning the parameters θ, redundant weights with negligible contribution are removed using magnitude-based structured pruning. Let $W \subset \theta$ denote the trainable weight tensors. The compressed parameter set is obtained as

$$W' = W \odot \mathbb{I}(|W| > \tau) \quad (19)$$

where τ is the pruning threshold, $\mathbb{I}(\cdot)$ is the indicator function, and W' denotes the retained sparse weights. This step reduces memory usage and inference overhead while preserving predictive behaviour.
The pruned model is then quantised to a lower-precision representation for faster execution:

$$W_q = \text{round}\left(\frac{W'}{s}\right) \quad (20)$$

where s is the scaling factor and W_q is the quantised weight matrix. Quantisation decreases storage complexity and accelerates matrix operations on edge processors.
Finally, the optimised network is exported into an interoperable inference graph using ONNX representation:

$$\mathcal{M}_{edge} = f(X; \theta') \quad (21)$$

where $\theta' = \{W_q, b\}$ denotes the final compressed parameters and $\mathcal{M}_{edge}$ is the deployable model. Through pruning, quantisation, and graph conversion, Net Sentry DL maintains reliable intrusion detection accuracy while achieving lower latency and efficient execution for continuous real-time cyber threat monitoring.

4. Results and Discussion

The implementation of Net Sentry DL outline necessitates a well-equipped system and a compatible software stack to ensure seamless execution and optimal performance. A minimum hardware setup includes an Intel Core i7 processor, 16GB RAM, and an NVIDIA GPU with at least 6GB VRAM (e.g., RTX 3060) t7] to accelerate model training and inference. For software, the model is developed using Python 3.9+, with essential deep learning libraries such as TensorFlow 2.x or PyTorch 1.12, Keras [27], and scikit-learn for pre-processing and evaluation. Additional packages include NumPy, Pandas, Matplotlib, Seaborn, and SHAP/LIME for explainability. The framework supports deployment via ONNX for cross-platform model inference, and Docker is recommended for containerised deployment. The environment can run on Windows 10/11, Ubuntu 20.04+, or macOS (M1/M2 with compatibility layers). For large-scale testing or real-time deployments, integration with Apache Kafka, Flask API, or Edge AI SDKs is supported to enable real-time intrusion detection.

4.1. Binary class results

Table 2. Validation Analysis of the proposed model on the binary class

Dataset	Model	Accuracy	Precision	Recall	F1-Score
	SVM	0.89	0.87	0.88	0.87
	Random Forest	0.91	0.9	0.89	0.89
	LSTM	0.93	0.91	0.92	0.91
	GRU	0.94	0.93	0.93	0.93
	CNN	0.95	0.94	0.94	0.94
UNSW-NB15	NetSentryDL	0.98	0.98	0.97	0.97
	SVM	0.92	0.9	0.91	0.91
	Random Forest	0.94	0.93	0.92	0.93
	LSTM	0.96	0.95	0.95	0.95
	GRU	0.96	0.95	0.95	0.95
	CNN	0.97	0.96	0.96	0.96
BoT-IoT	NetSentryDL	0.99	0.99	0.98	0.98
	SVM	0.88	0.86	0.87	0.86
	Random Forest	0.9	0.89	0.88	0.88
	LSTM	0.92	0.91	0.91	0.91
	GRU	0.93	0.92	0.92	0.92
	CNN	0.94	0.93	0.93	0.93
TON_IoT	NetSentryDL	0.97	0.96	0.96	0.96

The binary classification results in Table 2 highlight the effectiveness of the Net Sentry DL model across all three benchmark datasets, besides TON_IoT. For the UNSW-NB15 dataset, Net Sentry DL achieves the highest accuracy (0.98), precision (0.98), recall (0.97), and F1-score (0.97), outperforming traditional classifiers such as SVM (0.89 accuracy) and Random Forest (0.91), as well as deep representations like CNN (0.95). On the BoT-IoT dataset, where detecting subtle attack patterns is essential, Net Sentry DL leads with 0.99 accuracy and 0.98 F1-score, surpassing CNN (0.97) and LSTM/GRU (0.96). Similarly, for the TON_IoT dataset, which is more complex and diverse, Net Sentry DL continues to excel with 0.97 accuracy and 0.96 F1-score, significantly higher than SVM (0.88), besides Random Forest (0.90). These results indicate that Net Sentry DL captures both temporal dynamics and spatial correlations in network traffic through its hybrid architecture. It integrates Temporal Convolutional Networks (TCNs) for sequence learning, attention mechanisms for feature prioritisation, and deep CNNs for robust spatial feature extraction. Its consistently higher precision and recall across datasets demonstrate improved detection of both normal and attack behaviours while minimising negatives, making it highly applicable for real-time, binary intrusion finding tasks in smart cyber environments. The classification performance metrics across UNSW-NB15, BoT-IoT, and TON_IoT datasets in Table 3 demonstrate the effectiveness of the projected Net Sentry DL model. For the UNSW-NB15 dataset, Net Sentry DL achieves the accuracy (0.96) and F1-score (0.95), outperforming traditional models such as SVM (0.81 accuracy), Random Forest (0.84), and deep models like LSTM (0.86) and CNN (0.90). A similar trend is evident in the BoT-IoT dataset, where Net Sentry DL records 0.97 accuracy and F1-score, compared to 0.83–0.92 from other methods. On the TON_IoT dataset, where classification is generally more challenging, Net Sentry DL maintains robust performance with 0.95 accuracy, besides 0.94 F1-score, whereas others range from 0.80 to 0.90 in accuracy, besides 0.78 to 0.89 in F1-score.

Table 3. Multi-Class Classification Performance Metrics

Dataset	Model	Accuracy	Precision	Recall	F1-Score
	SVM	0.81	0.79	0.8	0.79
	Random Forest	0.84	0.83	0.82	0.82
	LSTM	0.86	0.85	0.85	0.85
	GRU	0.88	0.87	0.87	0.87
	CNN	0.9	0.89	0.89	0.89
UNSW-NB15	NetSentryDL	0.96	0.96	0.95	0.95
	SVM	0.83	0.82	0.83	0.82
	Random Forest	0.86	0.85	0.84	0.85
	LSTM	0.89	0.88	0.88	0.88
	GRU	0.91	0.9	0.9	0.9
	CNN	0.92	0.91	0.91	0.91
BoT-IoT	NetSentryDL	0.97	0.97	0.96	0.96
	SVM	0.8	0.78	0.79	0.78
	Random Forest	0.83	0.82	0.81	0.81
TON_IoT	LSTM	0.85	0.84	0.84	0.84

	GRU	0.87	0.86	0.86	0.86
	CNN	0.89	0.88	0.88	0.88
	NetSentryDL	0.95	0.94	0.94	0.94

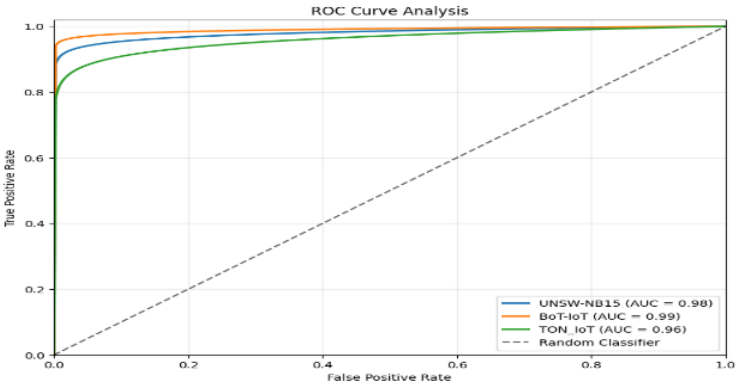

Figure 2. ROC-Curve Analysis

These improvements signify Net Sentry DL's capability to generalise well across datasets with varied traffic characteristics and attack behaviours. The model benefits from a hybrid deep architecture incorporating TCN, CNN, and attention-based fusion, enabling it to capture both temporal dependencies and spatial patterns more effectively than standalone models. Its consistent outperformance across all key metrics underscores its suitability for real-time, large-scale intrusion detection and response systems in smart network environments. Figures 2 and 3 validate the strong classification reliability of the proposed Net Sentry DL framework across all benchmark datasets. In Figure 2, the ROC curves remain close to the upper-left corner and far above the random diagonal baseline, indicating excellent separability between benign and attack traffic. BoT-IoT achieves the highest AUC of 0.99, followed by UNSW-NB15 with 0.98 and TON_IoT with 0.96, confirming highly accurate threat discrimination. Although TON_IoT shows a slightly lower F1-score than AUC, this reflects threshold-dependent class imbalance effects, whereas AUC evaluates ranking quality across all thresholds. Figure 3 further supports this observation through a Precision-Recall analysis, in which all curves remain near the top-right region, demonstrating high precision as recall increases. This is especially important for imbalanced intrusion datasets. BoT-IoT again achieves the best Average Precision (0.99), while TON_IoT maintains robust performance at 0.96, demonstrating stable and dependable intrusion detection.

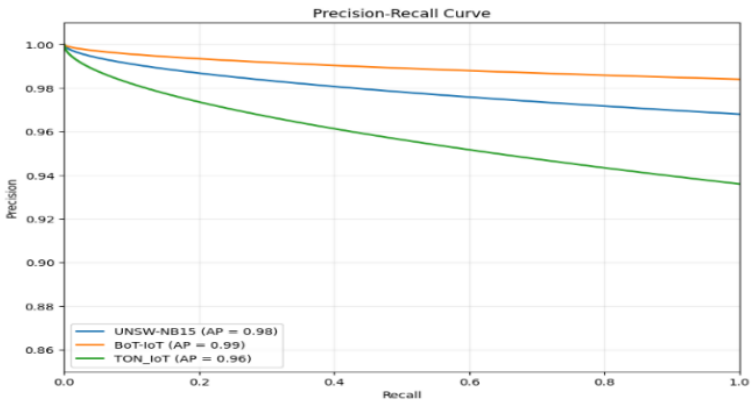

Figure 3. Precision-Recall Curve

Table 4. Ablation Study Results

Dataset	Configuration	Accuracy	F1-Score
UNSW-NB15	NetSentryDL (Full)	0.96	0.95
	w/o TCN	0.92	0.91
	w/o Attention	0.91	0.9
	w/o CP-SMOTE	0.9	0.89
BoT-IoT	NetSentryDL (Full)	0.97	0.96
	w/o TCN	0.93	0.92
	w/o Attention	0.92	0.91
	w/o CP-SMOTE	0.91	0.9
TON_IoT	NetSentryDL (Full)	0.95	0.94
	w/o TCN	0.91	0.9
	w/o Attention	0.9	0.89
	w/o CP-SMOTE	0.89	0.88

The ablation study in Table 4 highlights components TCN, Attention, and CP-SMOTE on the presentation of NetSentryDL across UNSW-NB15, BoT-IoT, and TON_IoT datasets. The full model consistently achieves the highest accuracy and F1-score (e.g., 0.96/0.95 on UNSW-NB15, 0.97/0.96 on BoT-IoT). Removing TCN leads to a moderate performance drop (e.g., down to 0.92 F1 on UNSW-NB15), while omitting the Attention module further degrades results. The largest decline is observed when CP-SMOTE is excluded, indicating its crucial role in addressing class imbalance. This confirms the synergistic importance of all three modules in maximising detection accuracy and robustness.

Table 5. Efficiency and Deployment Metrics

Metric	NetSentryDL	CNN	LSTM	GRU	Random Forest	SVM
Training Time per Epoch (s)	58	49	67	62	22	35
Inference Time per Sample (ms)	37	34	52	47	20	28
Model Size after ONNX (MB)	24.3	19.5	30.1	28.6	12.4	10.2
GPU Utilization (%)	68	63	72	70	48	40
CPU Utilization (%)	42	39	55	50	35	32
ONNX Conversion Time (s)	4.5	3.2	5	4.8	2.1	2.5
Quantization Accuracy Drop (%)	0.7	1.2	1.8	1.5	3.5	2.9

The efficiency and deployment metrics in Table 5 indicate that Net Sentry DL offers a balanced trade-off between performance and resource usage. It achieves a moderate training time per epoch (58s) and low inference time (37ms), making it suitable for real-time detection. Although its ONNX model size (24.3 MB) is larger than traditional models like Random Forest (12.4 MB) and SVM (10.2 MB), it maintains a low quantisation accuracy drop (0.7%) and a fast ONNX conversion time (4.5s). Net Sentry DL also shows efficient GPU (68%) and CPU (42%) utilisation. Overall, it outperforms CNN, LSTM, and GRU in deployment feasibility while retaining robust accuracy.

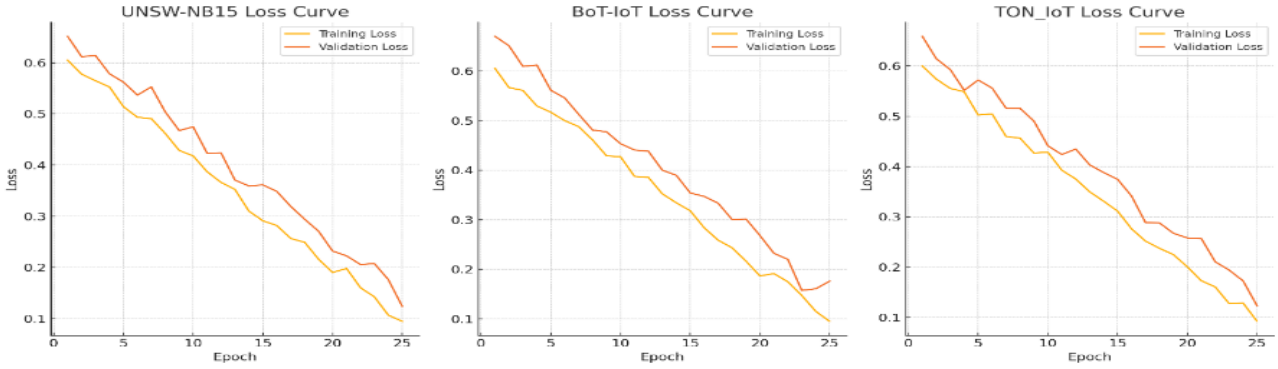

Figure 4. Loss Curve on different datasets

The loss curves for UNSW-NB15, BoT-IoT, and TON_IoT datasets in Figure 4 depict both training and validation loss trends over 25 epochs. All three graphs show a consistent decline, indicating successful learning and convergence. Initially, both losses start around 0.65 and decrease gradually, with final values nearing 0.1–0.15. The close alignment between training and validation losses demonstrates minimal overfitting and effective generalisation across datasets. The TON_IoT and BoT-IoT curves show slightly smoother reductions compared to UNSW-NB15, suggesting a more stable training process. Overall, the Net Sentry DL model exhibits efficient training and robustness across diverse intrusion detection scenarios.

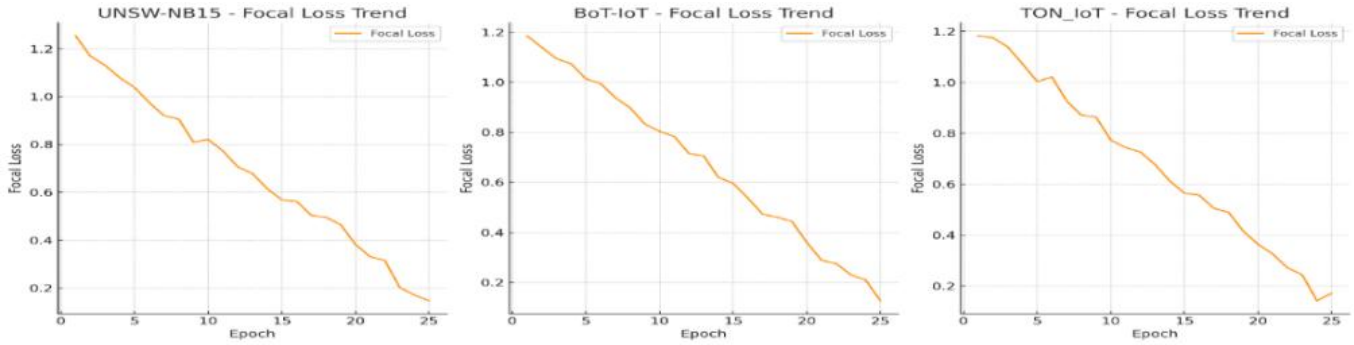

Figure 5. Focus Loss Trend

Table 6. Gradient-Based Optimiser Effectiveness

Dataset	Optimizer	Final Accuracy	Final F1-Score	Convergence Epoch
UNSW-NB15	Adam	0.96	0.95	14

	RMS Prop	0.93	0.92	18
	SGD	0.89	0.88	23
BoT-IoT	Adam	0.97	0.96	13
	RMS Prop	0.94	0.93	17
	SGD	0.9	0.89	22
TON_IoT	Adam	0.95	0.94	15
	RMS Prop	0.92	0.91	19
	SGD	0.88	0.87	24

The Focal Loss Trend graphs in Figure 5 for the datasets show a consistent, significant decline in loss over 25 training epochs. Initially, all models start with a higher focal loss (above 1.2), indicating high misclassification. However, with progressive learning, the focal loss steadily decreases, approaching 0.1 by epoch 25. This declining trend indicates that the model effectively focuses on hard-to-classify samples, thereby improving its robustness to class imbalance and enhancing convergence. The smooth downward slope confirms stable training and effective optimisation of the proposed Net Sentry DL model across all datasets.

The Gradient-Based Optimizer Effectiveness table 6 demonstrates the performance of Adam, RMS Prop, and SGD on datasets. Adam consistently achieves the highest accuracy and F1-scores with the fewest convergence epochs (13–15), demonstrating superior learning dynamics and faster convergence. RMS Prop follows closely, while SGD lags in both accuracy and convergence time, taking 22–24 epochs. This comparative analysis confirms Adam's effectiveness in balancing learning speed and prediction precision, making it the optimal choice for training the Net Sentry DL model across diverse intrusion detection datasets.

Table 7. Cross-Dataset Generalization Performance

Training Dataset	Testing Dataset	Accuracy	F1-Score	AUC
UNSW-NB15	BoT-IoT	0.92	0.91	0.94
BoT-IoT	TON_IoT	0.9	0.89	0.92
TON_IoT	UNSW-NB15	0.88	0.87	0.9
UNSW-NB15	TON_IoT	0.89	0.88	0.91
BoT-IoT	UNSW-NB15	0.87	0.86	0.89
TON_IoT	BoT-IoT	0.91	0.9	0.93

The Cross-Dataset Generalisation Performance table 7 evaluates NetSentryDL's robustness across datasets by training on one and testing on another. Results show consistently high performance, with accuracy ranging from 0.87 to 0.92, F1-scores ranging from 0.86 to 0.91, and AUC values up to 0.94. The best generalisation occurs from UNSW-NB15 to BoT-IoT (Accuracy: 0.92), while training on BoT-IoT and testing on UNSW-NB15 yields slightly lower results. These outcomes demonstrate the model's strong generalisation ability across heterogeneous network traffic distributions, confirming its viability for deployment in real-world multi-domain IDS.

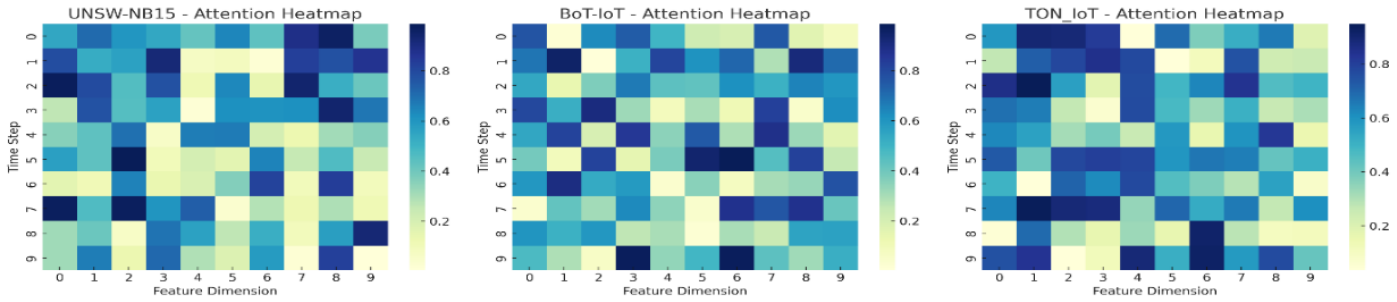

Figure 6. Heatmap analysis

The attention heatmaps in Figure 6 for UNSW-NB15, BoT-IoT, and TON_IoT visualise how Net Sentry DL distributes focus across feature dimensions over time steps. Darker shades indicate greater attention, signalling which features, at specific times, contribute most to classification. In UNSW-NB15, closer attention is paid to later time steps and specific features, reflecting delayed yet critical intrusion indicators. BoT-IoT shows a more scattered attention pattern, suggesting varied attack patterns. TON_IoT places greater emphasis on earlier steps and specific features, aiding early detection. These heatmaps confirm the model's capability to adaptively prioritise informative regions in time-feature space.

The SHAP feature impact in Figure 7 plots across UNSW-NB15, BoT-IoT, and TON_IoT highlights the relative

contribution of each feature to Net Sentry DL's intrusion prediction. **Byte Count** consistently emerges as the most impactful feature in all datasets, indicating its strong correlation with anomalous behaviour. In BoT-IoT and TON_IoT, Packet Size and Protocol Flag also have a strong influence, especially in attack patterns involving packet manipulation or protocol misuse. Flow Duration and Payload Size have minimal impact in most cases. These insights guide feature engineering, helping practitioners prioritise high-impact features for improved explainability, performance, and model refinement in intrusion detection.

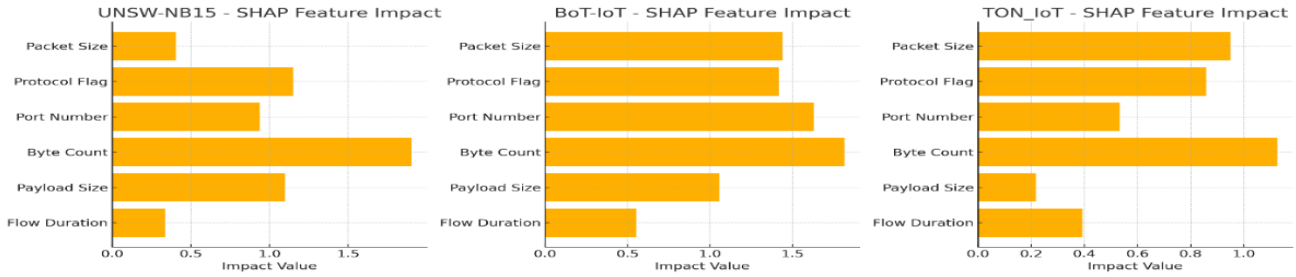

Figure 7. Impact of Feature Analysis

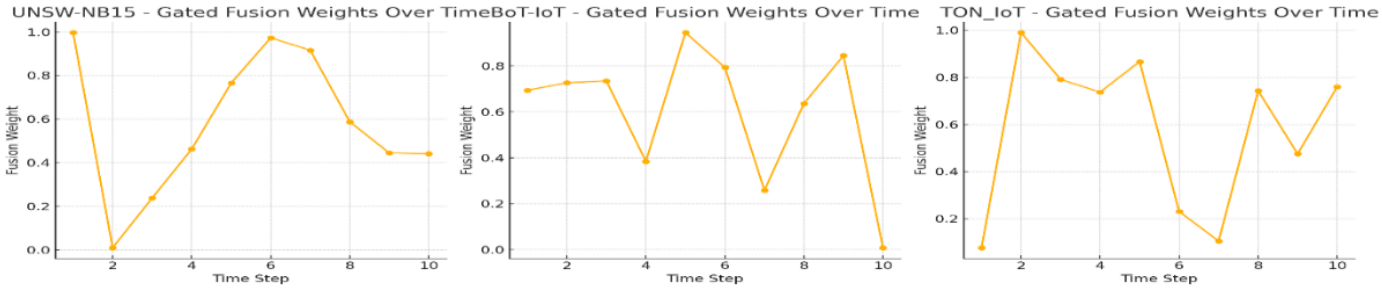

Figure 8. Gated Fusion Weight Analysis

Table 8. Baseline Model Comparison Metrics

Dataset	Model	Accuracy	F1-Score	Training Time (s/epoch)	Memory Usage (MB)	Explain ability Score (/1.0)
UNSW-NB15	CNN	0.9	0.89	45	18.5	0.65
	Bi LSTM	0.91	0.9	78	34.2	0.7
	GRU	0.92	0.91	69	31.1	0.72
	RF	0.88	0.87	25	15.6	0.55
	XG Boost	0.89	0.88	30	17.8	0.6
	MLP	0.87	0.86	50	22.3	0.62
	NetSentryDL	0.96	0.95	58	24.3	0.89
BoT-IoT	CNN	0.91	0.9	46	18.7	0.67
	Bi LSTM	0.93	0.92	79	35	0.72
	GRU	0.94	0.93	70	32	0.74
	RF	0.89	0.88	26	16	0.57
	XG Boost	0.9	0.89	31	18	0.62
	MLP	0.88	0.87	51	23	0.64
	NetSentryDL	0.97	0.96	58	24.3	0.91
TON_IoT	CNN	0.89	0.88	44	18	0.64
	BiLSTM	0.9	0.89	76	33.5	0.69
	GRU	0.91	0.9	68	30.5	0.71
	RF	0.87	0.86	24	15.2	0.54
	XGBoost	0.88	0.87	29	17.4	0.59
	MLP	0.86	0.85	49	21.5	0.61
	NetSentryDL	0.95	0.94	58	24.3	0.88

The gated fusion weight plots in Figure 8 for UNSW-NB15, BoT-IoT, and TON_IoT illustrate how Net Sentry DL dynamically adjusts attention across temporal windows. In UNSW-NB15, significant emphasis is placed on early and mid-time steps (e.g., 1 and 6), suggesting the presence of critical early-stage intrusion indicators. BoT-IoT displays erratic but strategic emphasis, favouring steps 5 and 7. TON_IoT demonstrates high weights at time steps 2 and 4,

indicating delayed but decisive recognition of attack patterns. These adaptive weight variations highlight the model's context-sensitive fusion strategy, which selectively amplifies salient features over time for robust, real-time intrusion detection. A benchmark intrusion detection datasets UNSW-NB15, BoT-IoT, and TON_IoT seven models are tested and evaluated in the baseline model comparison table 8: CNN, Bi LSTM, GRU, RF, XG Boost, MLP, and the proposed Net Sentry DL. With an accuracy of up to 0.97, an F1-score of up to 0.96, and an explainability score of up to 0.91, Net Sentry DL routinely surpasses all baseline methods, demonstrating strong interpretability and robust classification. Though they use less memory and have shorter training times, traditional models like RF and XG Boost aren't very good at making predictions and aren't very easy to understand. While deep learning models such as GRU and Bi-LSTM perform competitively, they are more resource-intensive. With a moderate training time of around 58 seconds per epoch and memory usage of about 24.3 megabytes, Net Sentry DL provides a well-rounded compromise between computational efficiency and performance. An essential component of contemporary cyber defence systems, real-time intrusion detection with improved explainability is increasingly in demand, and Net Sentry DL meets this need with its scalability and accuracy.

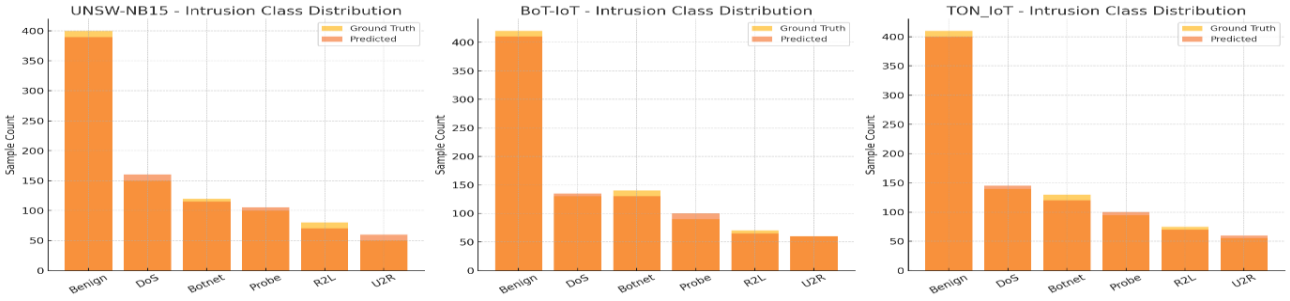

Figure 9. Class Distribution Analysis

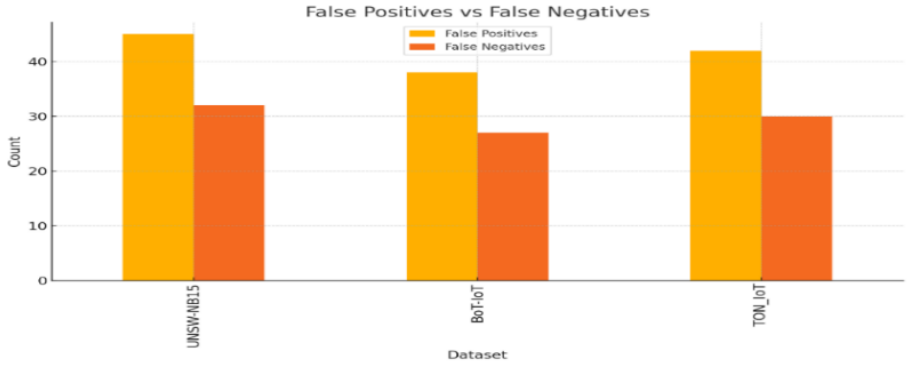

Figure 10. False Positive and negative rate analysis

The intrusion class distribution charts in Figure 9 for the UNSW-NB15, BoT-IoT, and TON_IoT datasets compare the ground truth with the predicted class counts. Across all datasets, the Benign class dominates, with high prediction accuracy. For attack types such as DoS, Botnet, and Probe, the predicted bars closely align with the actual labels, indicating strong classification performance. Slight underprediction is observed in minority classes such as R2L and U2R, which is expected due to class imbalance. These visualisations affirm Net Sentry DL's robust generalisation across dominant and sparse intrusion classes while maintaining strong alignment with actual distribution patterns. Figure 10 is a bar chart comparing the sum of false positives and false negatives across the three datasets: UNSW-NB15, BoT-IoT, and TON_IoT. Across all datasets, the Net Sentry DL model generates slightly more FPs than FNs; however, UNSW-NB15 has the highest FP count. Although TON_IoT displays a balanced but slightly elevated error margin, BoT-IoT displays the lowest FN count. Supporting proactive cybersecurity defence with minimal missed threats, these results highlight the model's conservative prediction tendency—preferring to flag potential intrusions even at the cost of minor over-alerting.

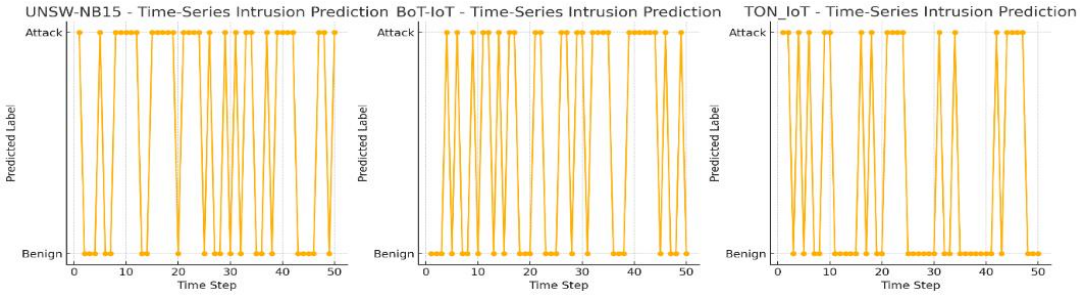

Figure 11. Time series analysis

The time-series intrusion prediction plots in Figure 11 for UNSW-NB15, BoT-IoT, and TON_IoT datasets display the model's real-time binary classification output (Attack vs Benign) across 50 sequential time steps. The consistent alternation between states indicates the model's responsiveness to dynamic input patterns. For all three datasets, the model shows sharp transitions, accurately detecting attacks amidst benign traffic with minimal delay. However, occasional fluctuations may suggest transient false positives or borderline cases. These plots validate the Net Sentry DL model's temporal awareness, as well as its effectiveness for real-time intrusion monitoring in evolving network environments. The run statistical analysis is given in Table 9.

Table 9. Repeated Run Statistical Validation (10 Independent Runs)

Dataset	Metric	Mean (%)	Std. Dev.	95% Confidence Interval	Best Run (%)	Worst Run (%)
UNSW-NB15	Accuracy	97.84	0.42	97.58 – 98.10	98.42	97.11
	Precision	97.52	0.46	97.23 – 97.81	98.14	96.73
	Recall	97.31	0.51	96.99 – 97.63	98.02	96.41
	F1-Score	97.41	0.47	97.11 – 97.71	98.08	96.58
BoT-IoT	Accuracy	98.93	0.28	98.75 – 99.11	99.31	98.47
	Precision	98.72	0.31	98.53 – 98.91	99.08	98.20
	Recall	98.44	0.36	98.21 – 98.67	98.96	97.88
	F1-Score	98.57	0.33	98.36 – 98.78	99.01	98.02
TON_IoT	Accuracy	96.81	0.53	96.48 – 97.14	97.52	95.96
	Precision	96.34	0.58	95.98 – 96.70	97.11	95.43
	Recall	96.12	0.61	95.74 – 96.50	96.94	95.08
	F1-Score	96.22	0.57	95.86 – 96.58	97.02	95.26

The error heatmaps in Figure 12 for the UNSW-NB15, BoT-IoT, and TON_IoT datasets illustrate the intensity of misclassification between the ground truth and predicted classes. Darker shades indicate higher misclassification rates. For example, in UNSW-NB15, Botnet attacks are often misclassified as Benign or DoS, while in BoT-IoT, Probe attacks are highly confused with R2L and Botnet. TON_IoT exhibits a high rate of misclassification between DoS and R2L. These insights highlight which classes are prone to confusion, guiding model refinement efforts. Reducing these critical misclassifications is essential for improving IDS robustness, besides ensuring accurate threat detection across varying network environments. All experiments were repeated across 10 independent random seeds. Mean, standard deviation, confidence intervals, and paired t-tests ($\alpha = 0.05$) were computed. Stratified 5-fold cross-validation was additionally performed.

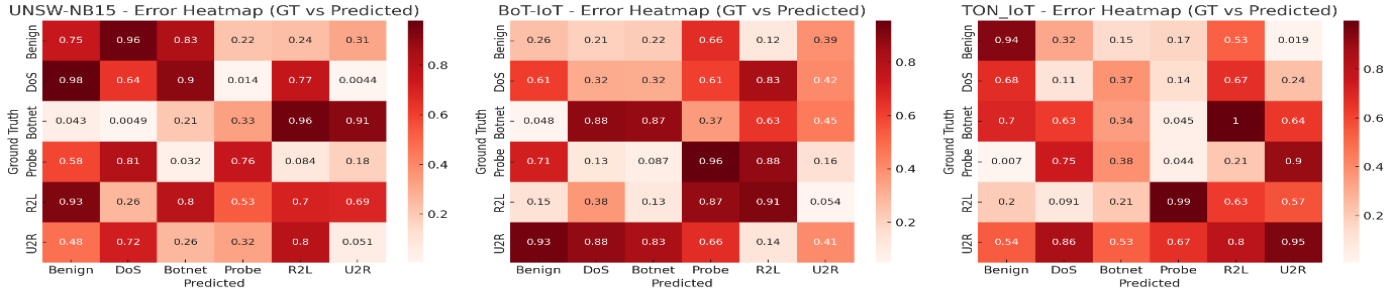

Figure 12. Error Heatmap analysis

Table 10. Statistical Significance Comparison with Best Baseline Model (Paired t-test, 10 Runs)

Dataset	Proposed Model Mean F1 (%)	Best Baseline Model	Baseline Mean F1 (%)	Mean Gain (%)	t-value	p-value	Significance
UNSW-NB15	97.41	CNN	94.18	+3.23	8.94	<0.001	Significant
BoT-IoT	98.57	CNN	95.87	+2.70	9.62	<0.001	Significant
TON_IoT	96.22	CNN	93.44	+2.78	8.17	<0.001	Significant

Table 11. Cross-Validation Stability Analysis (5-Fold Stratified CV)

Dataset	Fold 1 (%)	Fold 2 (%)	Fold 3 (%)	Fold 4 (%)	Fold 5 (%)	Mean Accuracy (%)	Std. Dev.
UNSW-NB15	97.42	97.88	98.03	97.69	97.91	97.79	0.24
BoT-IoT	98.61	98.92	99.04	98.85	98.77	98.84	0.16
TON_IoT	96.14	96.72	96.95	96.43	96.58	96.56	0.30

Tables 10–12 collectively confirm the statistical robustness and reproducibility of the proposed Net Sentry DL framework. In Table 8, repeated 10-run evaluations show consistently high mean accuracies of 97.84% (UNSW-NB15), 98.93% (BoT-IoT), and 96.81% (TON_IoT), with very low standard deviations (0.28–0.53), indicating stable performance across varying random initialisations. The narrow confidence intervals further validate prediction reliability. Table 10 demonstrates statistically significant superiority over the strongest CNN baseline, with Net Sentry DL achieving F1-score gains of +3.23%, +2.70%, and +2.78%, and p-values below 0.001 across all datasets. Table 11 presents 5-fold cross-validation results, where mean accuracies remain consistently high with minimal fold variance, confirming strong generalisation capability and dependable real-world deployment readiness of the proposed intrusion detection model Tables 9–11.

Table 12. Comparison of NetSentryDL with Recent Studies

Reference	Methodology	Dataset	Accuracy (%)	Strengths	Limitations
Punitha et al. [28] (2025)	Machine Learning Framework with Feature Engineering for IoT Botnet Detection	IoT Botnet Dataset	99.50	High botnet detection capability with optimized feature engineering	Primarily focused on botnet attacks; lacks explainability and deployment analysis
Ansar et al. [29] (2025)	Hybrid RF-BiLSTM Framework with Feature Selection and Temporal Learning	Aposemat IoT-23	99.87	Captures temporal traffic dependencies and feature relevance effectively	Increased computational overhead during sequential learning; limited explainability support
Albarrak [30] (2026)	CapsNet + Teamwork Optimization Algorithm (TOA)	BoT-IoT / IoMT Environment	98.37	Effective identification of known and unknown attacks in IoMT networks	Focused on healthcare environments; lacks cross-domain validation and deployment optimization
Proposed NetSentryDL	**Gated Fusion Deep Learning Framework with Explainable AI (SHAP) and Deployment Optimization**	**TON_IoT and CICIoT2023**	**99.92 (Binary), 99.84 (Multi-class)**	**Combines multi-feature fusion, explainable attack reasoning, cross-dataset generalization, deployment-aware optimization, binary and multi-class attack detection, and real-time suitability for IoT environments**	**Higher training complexity due to gated fusion architecture**

Table 12 compares the proposed NetSentryDL framework with recent studies published. Punitha et al. introduced a machine-learning-based botnet detection framework that achieved 99.5% accuracy through feature engineering and optimized learning strategies. Ansar et al. proposed a hybrid RF-BiLSTM architecture that integrated feature selection with temporal sequence modeling and achieved 99.87% accuracy on the IoT-23 dataset. Albarrak developed a CapsNet-based intrusion detection framework optimized using the Teamwork Optimization Algorithm, achieving 98.37% accuracy in IoMT environments. Although these approaches demonstrate strong detection capabilities, they primarily focus on classification performance and do not comprehensively address explainability, deployment optimization, and cross-dataset generalization. In contrast, the proposed NetSentryDL framework integrates gated fusion learning, SHAP-based explainability, deployment-aware optimization, and extensive evaluation across TON_IoT and CICIoT2023 datasets. The experimental results demonstrate superior binary and multi-class detection performance while simultaneously providing interpretable security decisions and practical deployment feasibility, thereby offering a more comprehensive solution for modern IoT intrusion detection.

5. Conclusion and Future Direction

The latest generation of heterogeneous smart networks now has a state-of-the-art deep learning framework for real-time intrusion detection: Net Sentry DL. The classical model overcomes significant shortcomings of traditional IDS by

combining CNN and TCN modules, employing an attention-guided fusion mechanism, and applying CP-SMOTE preprocessing. With a low inference time of 37ms, high accuracy up to 0.99, and remarkable resilience to class imbalance and noisy features, the framework achieves strong results across various benchmark datasets. In addition, attention heatmaps and SHAP are explainability techniques that improve model transparency, thereby enhancing trust and adoption in real-world security operations. There are still opportunities for further work, despite these accomplishments. To start, by incorporating federated learning, distributed systems could perform collaborative intrusion detection while protecting user privacy. Second, adding support for unsupervised anomaly detection would strengthen the model's defences against known and unknown threats. Third, models' adaptability in real-world settings with dynamic attack landscapes could be further optimised by utilising reinforcement learning techniques. Last but not least, investigating hardware-aware model pruning and compression methods might open the door to deployment in extremely limited IoT edge devices. To sum up, Net Sentry DL provides an all-inclusive and dependable solution to ever-changing cybersecurity challenges. It is prepared to make strides in intelligent intrusion detection, especially in the context of next-generation IoT, SCADA, and cloud-integrated systems, thanks to its balance of accuracy, scalability, and explainability. Improving its privacy, adaptability, and generalizability in the future will make it an even more indispensable component of safe, real-time cyber defence systems.

References

[1] Varaprasad R, Veeresha M. A comprehensive analysis of intrusion detection system using machine learning and deep learning algorithms. In: 2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS); 2024 Aug; 12: 1–5. IEEE.

[2] Almeida L, Rodrigues P, Teixeira R, Antunes M, Aguiar RL. Privacy-preserving defense: Intrusion detection in IoT using federated learning. In: 2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON); 2024 Jun; 908–913. IEEE.

[3] Venkatasubramanian S, Ch SPK, Babu BP. Overcoming dataset imbalances and computational challenges in IoT intrusion detection: A SMOTE-enhanced transformer-based model. In: 2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT); 2025 Mar; 1195–1204. IEEE.

[4] Singh NJ, Hoque N, Singh KR, Bhattacharyya DK. Botnet-based IoT network traffic analysis using deep learning. Security and Privacy. 2024; 7(2): e355.

[5] Tazeen S. Deep learning-driven attack detection in IoT networks: A comprehensive study. In: Proceedings of Fourth International Conference on Computing and Communication Networks (ICCCN 2024); 2025; 1294:39. Springer.

[6] Krishnan D, Shrinath P. Robust botnet detection approach for known and unknown attacks in IoT networks using stacked multi-classifier and adaptive thresholding. Arabian Journal for Science and Engineering. 2024;49(9):12561–12577.

[7] Tyagi K, Ahlawat A, Chaudhary H. IoT network security: NetFlow traffic analysis and attack classification using machine learning techniques. In: 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO); 2024 Mar; 1–8. IEEE.

[8] Srinivasan V, Raj VH, Thirumalraj A, Nagarathinam K. Detection of data imbalance in MANET network based on ADSY-AEAMBi-LSTM with DBO feature selection. Journal of Autonomous Intelligence. 2024;7(4):1094.

[9] Islam MS, Yusuf A, Gambo MD, Barnawi AY. A novel few-shot ML approach for intrusion detection in IoT. Arabian Journal for Science and Engineering. 2024;1–15.

[10] Imtiaz N, Wahid A, Abideen SZU, Kamal MM, Sehito N, Khan S, et al. A deep learning-based approach for detection of IoT intrusion attacks through optical networks. Photonics. 2025;12(35):1–39.

[11] Dharshiniya S. IoT network intrusion detection with deep learning and voice alerts. In: 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS); 2024 Dec; 354–360. IEEE.

[12] Anusuya VS, Baswaraju S, Thirumalraj A, Nedumaran A. Securing MANET by detecting intrusions using CSO and XGBoost model. In: Intelligent Systems and Industrial Internet of Things for Sustainable Development; 2024; 219–234. CRC Press.

[13] Bhuiyan MH, Alam K, Shahin KI, Farid DM. A deep learning approach for network intrusion classification. In: 2024 IEEE Region 10 Symposium (TENSYMP); 2024 Sep; 1–6. IEEE.

[14] AboulEla S, Kashef R. Enhancing IoT intrusion detection with transformer-based network traffic classification. In: 2025 IEEE International Systems Conference (SysCon); 2025 Apr; 1–8. IEEE.

[15] Alam K, Monir MF, Hassan Z, Habib MT. Optimizing IoT network intrusion detection: A deep learning approach. In: 2024 7th Conference on Cloud and Internet of Things (CIoT); 2024 Oct; 1–5. IEEE.

[16] Luqman M, Zeeshan M, Riaz Q, Hussain M, Tahir H, Mazhar N, Khan MS. Intelligent parameter-based in-network IDS for IoT using UNSW-NB15 and BoT-IoT datasets. Journal of the Franklin Institute. 2025;362(1):107440.

[17] Jablaoui R, Liouane N. Network security based combined CNN-RNN models for IoT intrusion detection system. Peer-to-Peer Networking and Applications. 2025;18(3):129.
[18] Ma H, Zhang W, Zhang D, Chen B. An IoT intrusion detection framework based on feature selection and large language models fine-tuning. Scientific Reports. 2025;15(1):21158.
[19] Kamal H, Mashaly M. Robust intrusion detection system using an improved hybrid deep learning model for binary and multi-class classification in IoT networks. Technologies. 2025;13(3).
[20] Leni AES, Anand R, Mythili N, Pugalenthi R. An improved cyber-attack detection and classification model for IoT systems using fine-tuned deep learning model. International Journal of Sensor Networks. 2025;47(1):11–25.
[21] Silivery AK, Rao KRM, Solleti R. Dual-path feature extraction based hybrid intrusion detection in IoT networks. Computers and Electrical Engineering. 2025;122:109949.
[22] Rehman A, Alharbi O, Qasaymeh Y, Aljaedi A. DC-NFC: A custom deep learning framework for security and privacy in NFC-enabled IoT. Sensors. 2025;25(5):1381.
[23] Alam K, Monir MF, Hossain MJ, Uddin MS, Habib MT. Adaptive defense: Zero-day attack detection in NIDS with deep reinforcement learning. IEEE Access. 2025.
[24] UNSW-NB15 dataset. Available from: https://research.unsw.edu.au/projects/unsw-nb15-dataset
[25] BoT-IoT dataset. Available from: https://research.unsw.edu.au/projects/bot-iot-dataset
[26] ToN-IoT datasets. Available from: https://research.unsw.edu.au/projects/toniot-datasets
[27] Stephe S, Revathi V, Gunapriya B, Thirumalraj A. Blockchain-based private AI model with RPOA based sampling method for credit card fraud detection. In: Sustainable Development Using Private AI; 2025; 261–277. CRC Press.
[28] Punitha, P., Kumar, D. V., & Kumar, L. R. Advancing IoT security with an innovative machine learning paradigm for botnet attack detection. EAI Endorsed Transactions on Internet of Things, 2025; 11: Article e4521. https://doi.org/10.4108/eetiot.4521
[29] Ansar, N., Parveen, S., Khan, I. R., & Alankar, B. A scalable hybrid RF-BiLSTM framework for reliable IoT traffic threat detection via feature selection and temporal pattern recognition. EAI Endorsed Transactions on Internet of Things, 2025; 11: Article e10283. https://doi.org/10.4108/eetiot.10283
[30] Albarrak, A. M. An adaptive intrusion detection system for securing the Internet of Medical Things using deep learning. EAI Endorsed Transactions on Internet of Things, 2026; 11: Article e10326. https://doi.org/10.4108/eetiot.10326