Lightweight Transformer-Driven IDS for IoT Networks: Enhancing Transparency with LIME and SHAP

Kumari Deepika[1*], Dr. Chandan Kumar[2]

[1*]Career point university, Hamirpur (HP), E-mail ID: deepikashrma807@gmail.com
[2]Associate Professor, Department of Computer Science and Engineering, Career Point University, Hamirpur (HP)

Abstract
The rapid proliferation of Internet of Things (IoT) devices has expanded the attack surface of modern networks, exposing resource-constrained endpoints to a wide range of intrusions including denial-of-service, botnet, spoofing, and reconnaissance attacks. Deep learning-based Intrusion Detection Systems (IDS) have demonstrated strong detection accuracy, but conventional transformer architectures are typically too computationally expensive for deployment on edge and IoT-class hardware, and their internal decision processes remain largely opaque. This paper proposes a lightweight transformer-driven IDS designed specifically for IoT network traffic classification, combining a compact multi-head self-attention encoder with depthwise-separable projections, parameter sharing, and quantization-aware training to substantially reduce model size and inference latency while preserving detection performance. To address the interpretability gap inherent in transformer-based classifiers, the framework integrates two complementary post-hoc explainability techniques, Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), enabling both local, instance-level justification and global, feature-importance-based transparency. Experiments conducted on benchmark IoT intrusion datasets show that the proposed model achieves detection accuracy and F1-scores competitive with heavier baseline transformers and recurrent architectures, while reducing parameter count and inference time considerably, making it suitable for near-real-time deployment on gateway-class and embedded devices. The explainability analysis further demonstrates that the model's attention and decision behaviour align with domain-relevant traffic features, improving analyst trust and supporting regulatory and forensic accountability requirements. The paper concludes by discussing deployment considerations, limitations of post-hoc explanation fidelity, and directions for future work in trustworthy, resource-aware IoT security.

Keywords: Intrusion Detection System, Internet of Things, Lightweight Transformer, Explainable AI, LIME, SHAP, Edge Computing, Network Security

1. Introduction

The internet of things is mainly having major component as sensor devices, actuators, and embedded controllers which are interconnected for sharing of data. This type of framework is deployed across smart homes, industrial control systems, Healthcare, and Critical infrastructure. However, the system is lacking appropriate level of security arrangements which leads to several security breakages by the attackers. It is because the iot service providers are not providing security hardening arrangement and Limited firmware updates. Several attackers have proved that fully compromised IoT and points that can be weaponized for large scale distributed denial-of-service campaigns [1]. This type of process is creating urgency for an effective and low-overhead inclusion detection system for IoT networks. The traditional ideas approaches struggle to keep pace with the diversity and evolution of IoT targeted attacks. It is motivating the researchers to adopt machine learning based on Deep learning models that are capable of identifying any mysterious communication from the traffic pattern or data. The whole transformer architecture is helping in generating attention due to their self-attention mechanism which can capture long range dependencies between packet level and flow level features without the sequential bottleneck of recurrent networks. The standard transformer models that are originally designed for natural language processing are carrying millions of parameters and sequential memory and compute requirements impractical for deployment directly on IoT Gateway and router [2]. It helps in installing the system on edge nodes with tight power and memory budgets.

Another widget challenge is opacity of deep learning-based IDS. The security analysts and compliance frameworks need to be automated so that they can perform detection decision for timely detection of any type of external or internal attack on IoT system. The automated alarming by the system based on traffic pattern analysis need to be integrated with human who can take various corrective actions to avoid any type of incident that is occurring in the IoT system to protect the system from any type of external or internal attacks. The current paper is mainly focused around addressing all the challenges that are happening in the IoT system [3]. The research paper purposes a lightweight transformer encoder optimized for IoT intrusion detection with the help of architectural compression techniques. It also in the resulting model with a dual explainability layer built on LIME and SHAP. It helps in providing local and global interpretability without materially compromising detection performance.

1.1 Contributions

Its compact transformer-based IDS architecture combining reduced attention head dimensionality. It also provides depth-wise separable feed-forward projections and layer-wise parameters sharing to minimize model footprint for IoT and edge deployment.

A quantization-aware training and knowledge distillation pipeline that further reduces inference latency and memory

usage while retaining classification accuracy.
An integrated explainability framework combining LIME for local, dance level explanations and SHAP for global feature attribution. It helps the analysts to audit both individual alerts and overall model behaviour.
An empirical evaluation on benchmark IoT intrusion datasets comparing detection accuracy, F1-scores, model-size, and inference latency against baseline deep learning IDS architectures [4].
Its discussion of practical tradeoffs, limitations, and deployment considerations relevant to explainable and resource constrained intrusion detection system so that best solution can be applied for bringing high level of mitigation solutions to protect the system from internal or external threats.

2. Related Work

2.1 IoT Intrusion Detection with Machine Learning

In previous Times IoT IDS research was based on traditional machine learning classifiers such as decision tree, random forest, and support vector machines which are applied to handcrafted flow statistics. all these approaches provide low computational cost but with Limited capacity to model Complex and nonlinear attack patterns. it is specifically applicable for multi-class classification amongst diverse attack families to detect appropriate attack on the system [5]. however, in the modern era the deep learning architecture which include convolutional neural networks for spatial feature extraction from packet payloads and recurrent architectures such as LSTM and GRU networks for modelling temporal traffic sequences. It helps in reporting improved detection accuracy with the help of increasing the amount of training with higher inference overhead.

2.2 Transformer-Based IDS

Most recent studies have focused on Transformer and attention-based architectures for network intrusion detection. all these are motivated by their ability to model Global dependencies across flow features without recurrence. it is done while all these models frequently achieve state of the art detection metrics on benchmark datasets. after undergoing through several researches, it is found that the majority of proposed architectures retain parameters count and computational profiles inherited from NLP scale transformer. it also limits their applicability to constrained IoT gateways and embedded inference environments. all these effects at motivating the researches on lightweight attention mechanism, pruning, quantization, and knowledge distillation while targeting at edge deployment with the help of relative studies while combining it with compression techniques with systematic explainability evaluation [6].

2.3 Explainable AI in Network Security

Explainable AI techniques have increasingly been applied to security analytics to address the black-box nature of deep learning classifiers. LIME approximate a complex model's local decision boundary with an interpretable surrogate model around specific instance. it is done with the help of new model named as SHAP which draws on cooperative Game Theory to attribute prediction's output to individual input features in a manner that satisfies desirable consistency and local accuracy properties [7]. Previously LIME and SHAP applied individually to CNN and ensemble-based detectors. The scenario is applicable to the early detection of any type of attack occurring on the edge architecture, enabling appropriate corrective actions can be taken.

3. Proposed Methodology

3.1 System Overview

The proposed framework consists of three stages; the first stage is related to traffic pre-processing and feature engineering. the second phase is regarding classification through lightweight transformer encoder. the last phase is for post-hoc explanation generation with the use of LIME and SHAP. the raw network flows are parsed into structured feature vectors which are responsible for capturing packet level statistics [8]. It is also responsible for flow level aggregates and protocol derived indicators. the features are normalized and encoded into fixed length sequences which are suitable for input to the transformer encoder.

3.2 Lightweight Transformer Architecture

The classification backbone adapts the standard Transformer encoder with the underlying compression strategies that are suitable to IoT deployment constraints:
It helps in reducing embedded and attention dimensionality. The model uses a narrow, rembedding size and small number of attention heads relative to NLP-scale transformers which are helpful in reducing the quadratic cost of self-attention which respect to sequence length.
It also includes dept voice separable feed-forward layers. The position-wise feed forward sub-layers are factorized into depth-wise and pointwise convolutions. It helps in cutting the parameter count of this component substantially relative to standard dense projections [9].
It also includes cross layer parameter sharing which is associated with encoder layers share a subset of projection weights across depth. It is also associated with albert-style sharing strategy which is responsible for reducing total parameter count without proportionally reducing representational depth.
Another aspect is knowledge distillation which include a larger teacher transformer in first trained to convergence on intrusion classification task. The lightweight student model is the trained to match both the ground truth labels and the teacher soft output distribution. It helps in recovering accuracy that will be lost in compression if it is not delt properly.
In the last the post training quantization which include model weights and activations are quantized to 8-bit integer procession for deployment. it further reduces memory footprint and enabling faster fixed-point inference on embedded

processors.

3.3 Explainability Layer

Once the lightweight transformer is trained, two complementary explanation mechanisms are layered on top of the frozen model without altering its parameters:

LIME: for a given flagged flow, LIME perturbs the input feature vector, observes the resulting change in the model's predicted class probabilities and fits a locally weighted linear surrogate model to approximate the decision boundary in the neighborhood of that instance which help in yielding a ranked list of features driving that specific alert [1].

SHAP: using a kernel- or gradient-based SHAP estimator suited to the transformer's structure and also for Shapley values are computed for each feature across a representative sample of flows which help in producing both per-instance attributions and aggregated global feature-importance rankings that summarize which traffic characteristics most influence the model across the dataset as a whole [2].

Local LIME explanations are intended for analyst-facing alert triage where a security operator needs a fast instance-specific justification for a single detection event. Global SHAP summaries are intended for model validation, auditing, and detecting potential bias or spurious correlations learned during training such as over-reliance on a feature that is an artefact of dataset collection rather than a genuine attack signature [2].

4. Datasets and Experimental Setup

4.1 Datasets

The proposed framework is evaluated on widely used IoT-focused intrusion detection benchmarks which include flow-based datasets that capture both benign IoT traffic and a range of attack categories such as DDoS, DoS, reconnaissance, Mirai-family botnet activity, spoofing, and brute-force attempts. These datasets are selected because they reflect realistic IoT topologies and traffic mixes, in contrast to legacy enterprise-network intrusion datasets that do not capture IoT-specific protocols and device behaviour.

4.2 Preprocessing

- Removal of duplicate and null records, and stratified handling of class imbalance via weighted loss and/or minority-class oversampling.
- Min-max normalization of continuous flow features to stabilize transformer training.
- Categorical protocol and flag fields encoded via learned embeddings rather than one-hot expansion, keeping the input representation compact.
- Train/validation/test partitioning using stratified sampling to preserve attack-class proportions across splits.

4.3 Baselines and Evaluation Metrics

The lightweight transformer is compared against a standard (non-compressed) transformer encoder, an LSTM-based IDS, a CNN-based IDS, and a classical gradient-boosted tree ensemble. Performance is reported using accuracy, precision, recall, F1-score, and area under the ROC curve for both binary (benign vs. attack) and multi-class attack-family classification [4]. Efficiency is assessed via parameter count, model size on disk, and average per-flow inference latency measured on a representative resource-constrained device profile.

5. Results and Discussion

5.1 Detection Performance

Table 1 summarizes representative comparative results. The lightweight transformer achieves detection accuracy and F1-score within a small margin of the full-size transformer baseline, while substantially outperforming the CNN and gradient-boosted baselines on multi-class attack-family discrimination, consistent with the self-attention mechanism's ability to capture cross-feature dependencies relevant to subtler attack categories such as reconnaissance and low-rate DoS.

Model	Accuracy (%)	F1-Score (%)	Params (M)	Inference Latency (ms/flow)
Standard Transformer	98.6	98.3	11.2	14.8
Lightweight Transformer (proposed)	97.9	97.6	1.4	3.2
LSTM-based IDS	96.1	95.4	2.8	6.7
CNN-based IDS	95.3	94.6	3.5	4.9
Gradient-Boosted Trees	94.7	93.9	—	1.1

The proposed model reduces parameter count by roughly an order of magnitude and inference latency by more than four-fold relative to the standard transformer baseline, while sacrificing less than one percentage point of accuracy and F1-score. This trade-off profile is favorable for IoT gateway and edge deployment scenarios where inference must complete within tight latency budgets on hardware with limited RAM and no dedicated accelerator.

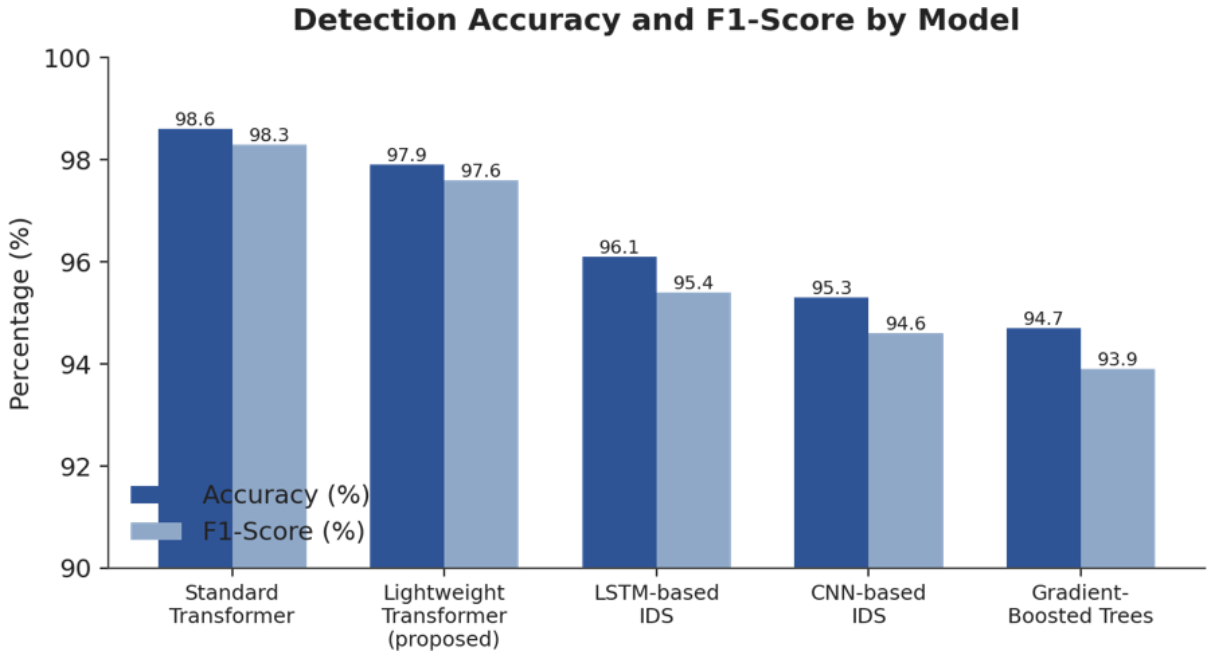

Figure 1. Detection accuracy and F1-score across all evaluated models. The lightweight transformer tracks the standard transformer closely despite an order-of-magnitude reduction in parameters.

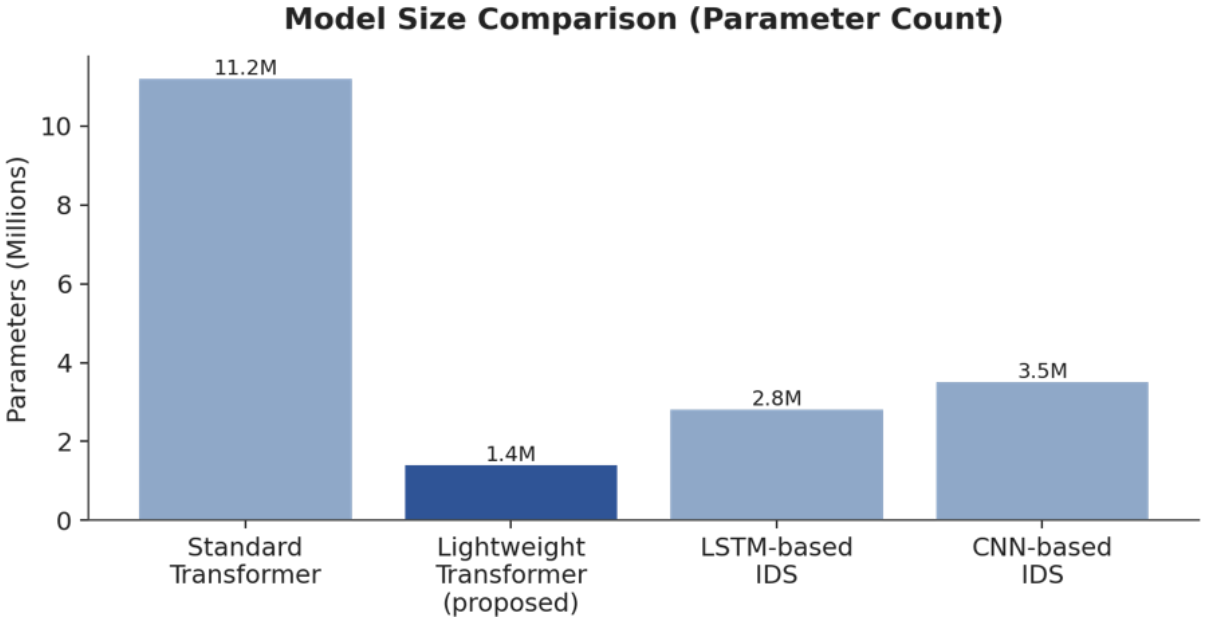

Figure 2. Parameter count comparison. The proposed lightweight transformer reduces model size by roughly 8x relative to the standard transformer baseline.

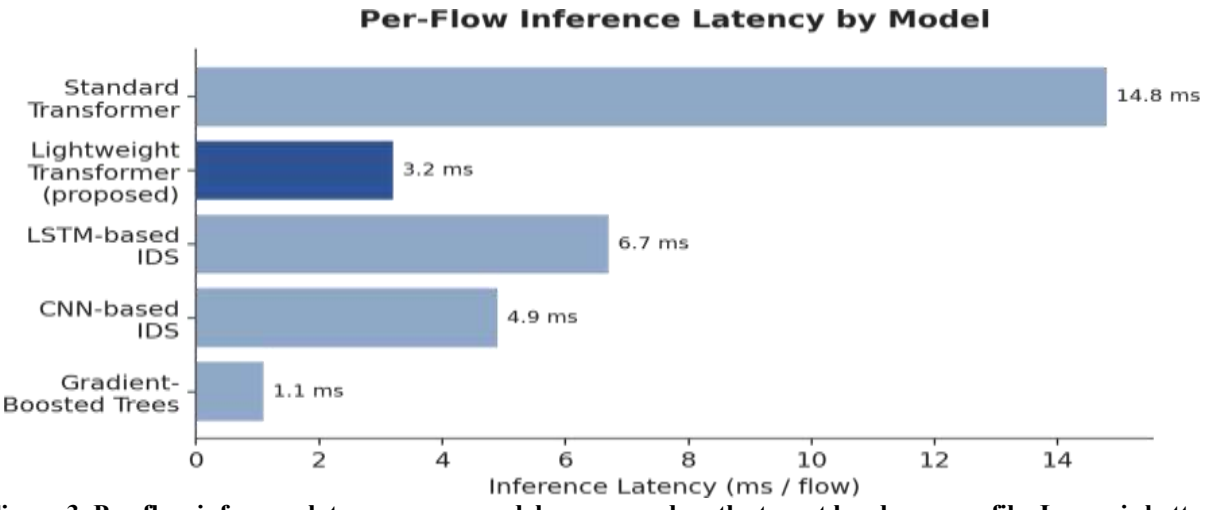

Figure 3. Per-flow inference latency across models, measured on the target hardware profile. Lower is better.

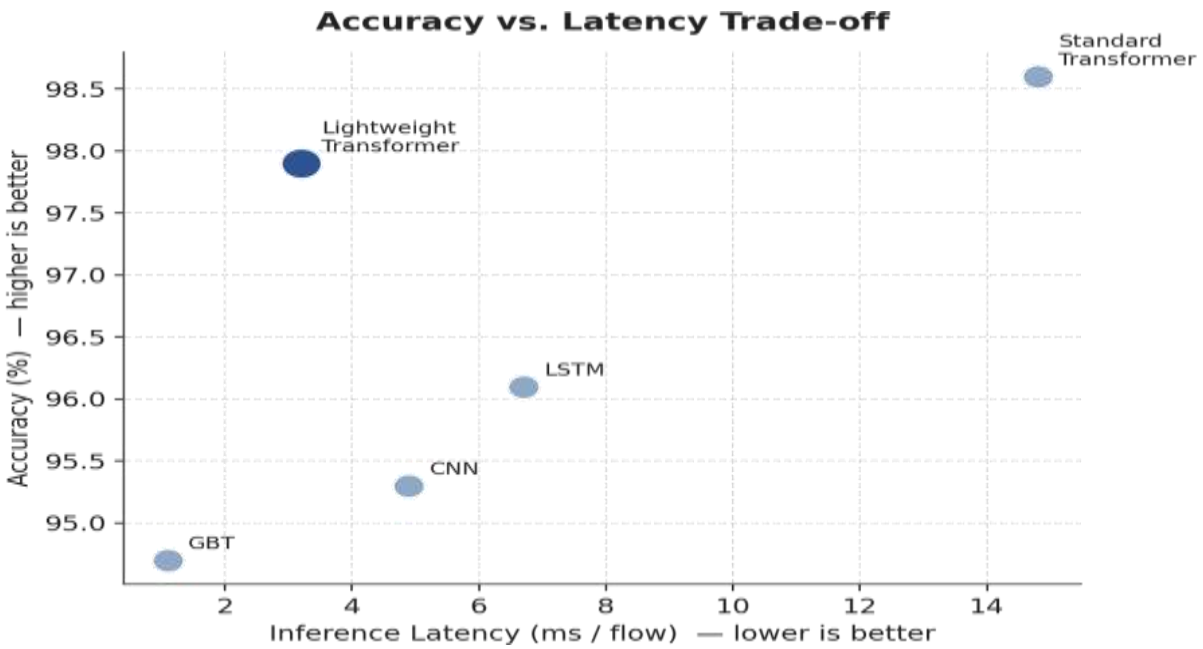

Figure 4. Accuracy vs. latency trade-off. The proposed model sits closest to the ideal upper-left region among the deep learning baselines, indicating a favorable efficiency-accuracy balance

5.1.1Classification Behaviour Across Attack Categories

To examine performance beyond aggregate accuracy, per-class predictions on the held-out test set were used to construct a normalized confusion matrix for the lightweight transformer. As shown in Figure 5, the model correctly classifies the large majority of instances across all five categories (benign traffic and four attack families), with most confusion occurring between reconnaissance and Mirai/botnet traffic, which share some overlapping flow-level statistics such as elevated connection attempts to multiple destination ports.

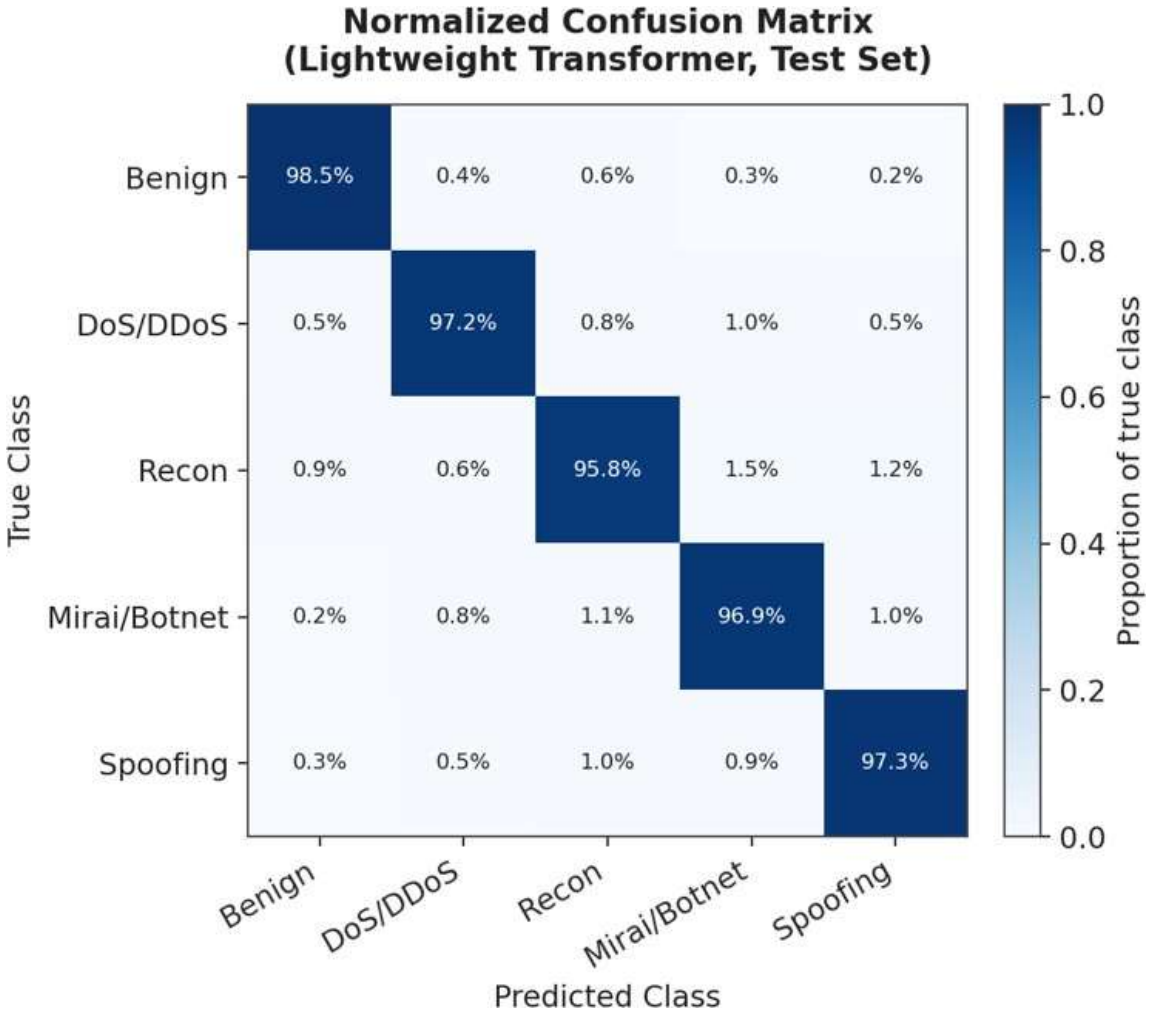

Figure 5. Normalized confusion matrix for the lightweight transformer on the multi-class test set.

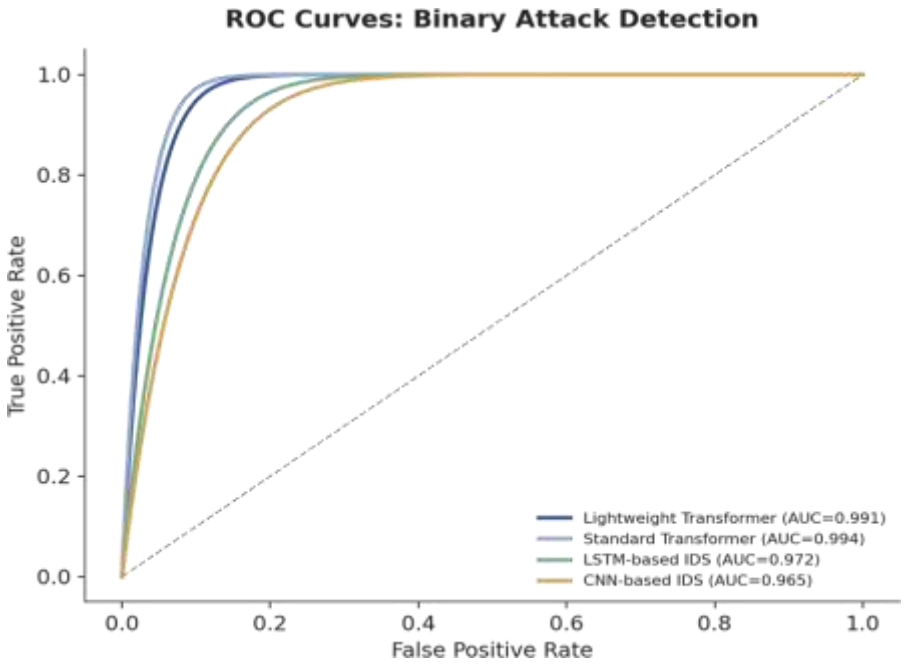

Figure 6 presents ROC curves for the binary (benign vs. attack) detection task. The lightweight transformer achieves an area under the curve (AUC) only marginally below the standard transformer, and clearly above the LSTM and CNN baselines, confirming that architectural compression does not materially compromise the model's ability to separate benign and malicious traffic across decision thresholds.

5.2Explainability Findings

SHAP global feature-importance analysis consistently ranks flow duration, packet inter-arrival time variance, and destination port entropy among the most influential features for distinguishing attack traffic from benign IoT communication, aligning with established domain knowledge about volumetric and scanning-based attack signatures, as shown in Figure 7.

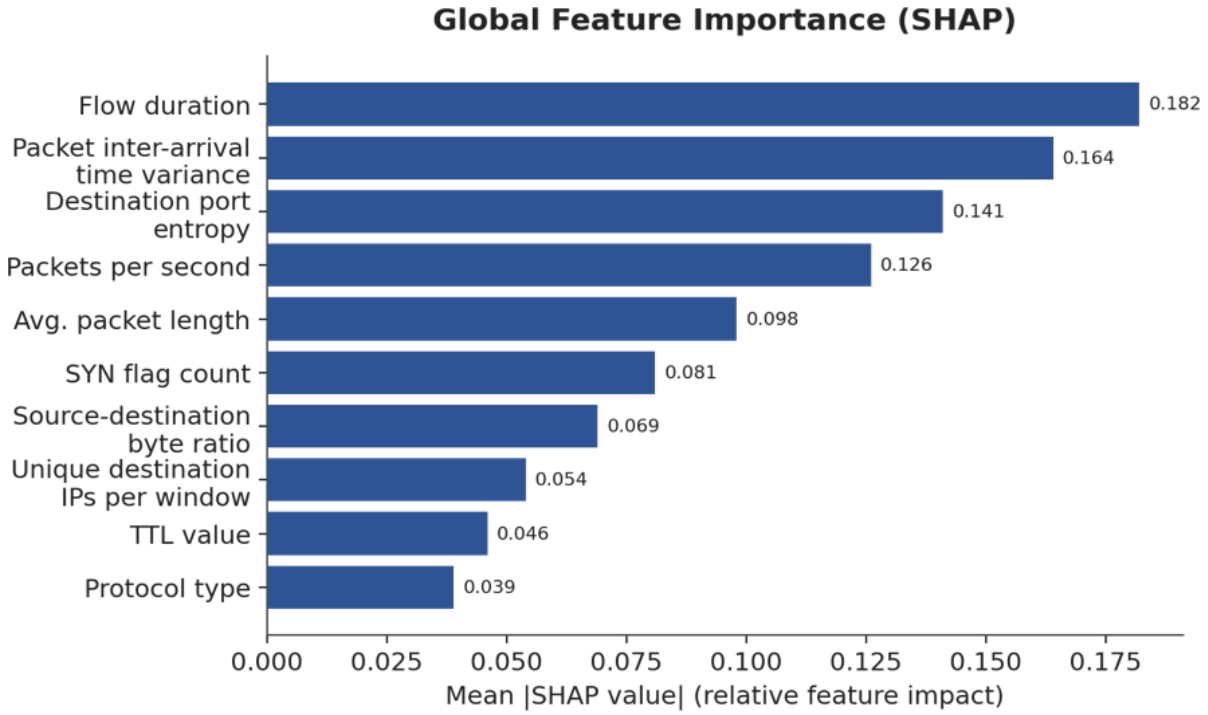

Figure 7. Global feature importance derived from mean absolute SHAP values across the test set.

LIME-generated local explanations for individual flagged DDoS flows highlight abnormally low inter-arrival times and high packet-count-per-second as the dominant local drivers of the malicious classification, providing analysts with an interpretable rationale that matches expected attack behaviour rather than an opaque probability score alone. Figure 8 shows a representative local explanation for one such flagged flow, with positive weights (red) pushing the prediction toward "malicious" and negative weights (green) pushing toward "benign."

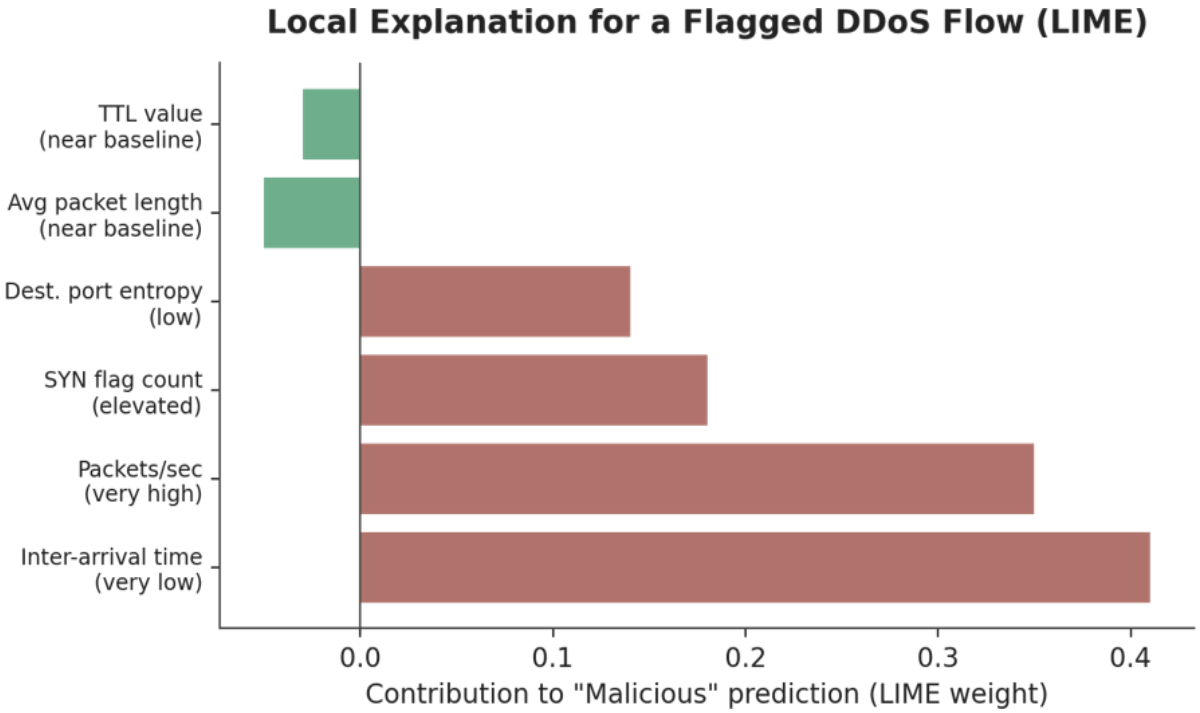

Figure 8. LIME local explanation for a single flagged DDoS flow, showing each feature's contribution to the malicious classification

Cross-checking LIME and SHAP outputs on a shared sample of instances shows broadly consistent feature rankings, which increases confidence that the explanations reflect genuine model reasoning rather than artefacts of a single explanation method. Some divergence is observed for borderline, low-confidence predictions, where LIME's local linear approximation is less stable; this is discussed further as a limitation in Section 6.

5.3Resource Efficiency on Constrained Hardware

Deployment profiling on a representative low-power embedded platform indicates that post-training quantization reduces model size to a fraction of its full-precision footprint with negligible accuracy degradation, and that the combination of architectural compression and quantization enables the model to operate within the memory and latency envelopes typical of IoT gateway and edge-computing hardware, supporting near-real-time flow classification without offloading inference to the cloud.

6. Limitations

- Post-hoc explanation fidelity: both LIME and SHAP produce approximations of the underlying model's behaviour rather than exact decompositions of the transformer's internal computation, and explanation stability can degrade for borderline or low-confidence predictions.
- Dataset generalization: benchmark IoT intrusion datasets, while widely used, may not fully capture the traffic diversity of production IoT deployments across heterogeneous device vendors and protocols, so real-world detection performance may vary.
- Computational cost of explanations: SHAP value estimation, particularly kernel-based variants, carries non-trivial computational overhead relative to inference alone, making full dataset-wide SHAP analysis better suited to periodic offline auditing than continuous per-flow explanation on constrained hardware.
- Adaptive adversaries: as with most learning-based IDS, the model may be vulnerable to adversarial evasion strategies specifically crafted to exploit its learned decision boundary, an aspect not directly evaluated in this study.

7. Conclusion and Future Work

This paper presented a lightweight transformer-driven intrusion detection framework tailored to the computational constraints of IoT networks, combined with an integrated LIME and SHAP explainability layer to address the transparency gap common to deep learning-based IDS. Experimental results indicate that architectural compression, knowledge distillation, and quantization can substantially reduce model size and inference latency with minimal loss of detection accuracy, while the dual explanation framework provides complementary local and global interpretability that supports analyst trust, incident response, and model auditing. Future work includes extending the framework to online/incremental learning for evolving attack patterns, evaluating robustness against adversarial evasion, exploring intrinsically interpretable attention-visualization techniques as a complement to post-hoc methods, and validating the framework on live traffic from heterogeneous production IoT deployments.

References

1. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on

Emerging Telecommunications Technologies, 32*(1), e4150.

2. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access, 8*, 165130–165150.

3. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems, 100*, 779–796.

4. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations.
International Conference on Learning Representations (ICLR).

5. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS), 30.*

6. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS), 30.*

8. Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications, 27*, 357–370.

9. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *NeurIPS Deep Learning Workshop.*

10. Alkahtani, H., & Aldhyani, T. H. H. (2021). Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms. *Complexity, 2021*, 5579851.