LEMD-GARF: Lightweight Edge-based Malware Detection Using Genetic Algorithm Feature Selection and Random Forest Classification

Ali Rezazadeh
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
ali_rezazadeh@elec.iust.ac.ir

Majid Nezarat
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
majid_nezarat@elec.iust.ac.ir

Ali Kalate
Faculty of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
a.kalateh@sbu.ac.ir

Hadi Shahriar Shahhosseini
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
shahhoseini@iust.ac.ir

Amin Rezaei
Department of Computer Engineering & Computer Science
California State University Long Beach
Long Beach, CA, USA
amin.rezaei@csulb.edu

***Abstract*—With the rapid expansion of the Internet of Things (IoT) and the growing reliance on edge computing, edge devices have become prime targets for cyber attackers. Given their critical role in modern network infrastructures, ensuring edge device cybersecurity is no longer optional but essential. Many adversaries attempt to penetrate the network core by compromising edge nodes, while others seek to disrupt localized services. In both cases, early detection of cyberattacks is crucial to protect users and prevent malicious activities. Traffic analysis is among the most effective approaches for mitigating zero-day attacks, as it enables the identification of unknown anomalous behaviors. In this paper, we present a lightweight a machine learning-based traffic analyzer capable of detecting malicious traffic in real time. By utilizing a Random Forest (RF) algorithm combined with Genetic Algorithm (GA)-based feature selection, the proposed system achieves 98.19% detection accuracy while maintaining computational efficiency suitable for resource-constrained edge environments. Experimental results demonstrate that the system can process each traffic sample within 0.189 microseconds, enabling ultra-fast and low-power detection. The combination of low energy consumption, high accuracy, and real-time performance makes the proposed solution an excellent candidate for deployment in next-generation edge devices.**

***Index Terms*—IoT Security; Traffic Analysis; Real-Time Detection; Machine Learning; Computational Efficiency**

I. Introduction

The rapid evolution of network technologies has made the Internet an integral part of modern life, while simultaneously introducing increasingly sophisticated cybersecurity threats [1], [2]. These challenges are amplified by the rise of the Internet of Things (IoT), where billions of resource-constrained devices expand the attack surface and generate massive data volumes [3]–[5]. Ensuring security in IoT networks is critical, but limited device capabilities hinder complex safeguards, making resource-aware designs essential [6], [7].

The rapid evolution of IoT-focused botnets and malware has enabled malicious traffic to blend into legitimate communication, facilitating large-scale attacks such as Distributed Denial of Service (DDoS), data exfiltration, and self-propagating campaigns [8], [9]. Centralized cloud-based detection approaches suffer from latency, privacy, and bandwidth limitations, reducing their effectiveness for timely mitigation [10]. In contrast, edge-based traffic analysis enables low-latency detection while accommodating the resource constraints of IoT devices, allowing early identification and containment of threats before widespread propagation [11]–[13].

In this paper, we propose **LEMD-GARF**, a Lightweight Edge-based Malware Detection using Genetic Algorithm feature selection and Random Forest classification. To address the challenges of high-dimensional traffic data, we first employ a Genetic Algorithm (GA) for feature selection, which reduces computational complexity while preserving the most discriminative attributes. On top of this optimized feature space, a Random Forest (RF) classifier is deployed to achieve accurate and robust detection of malicious traffic with minimal overhead. Unlike many existing approaches that remain purely software-based, our solution is implemented directly on hardware, enabling real-time operation with low power consumption. By combining feature reduction, machine learning, and hardware acceleration, the proposed system demonstrates the potential of delivering scalable, practical, and edge-friendly malware traffic detection for next-generation IoT networks. The contributions of this paper are as follows:

- Proposing a hardware-efficient malware detection framework for low-power edge environments. The system integrates feature reduction, machine learning, and hardware acceleration, enabling on-device, real-time operation;
- Employing a GA-based feature selection mechanism to

identify the most discriminative subset of flow-level features, reducing computational complexity by nearly 70% while preserving high detection accuracy;
- Implementing and verifying a hardware prototype of the RF classifier on an FPGA platform, achieving 98.19% detection accuracy with a 0.189 µs inference latency and ultra-low power consumption.

II. Background and Motivation

In this section, we review related work, discuss existing research gaps, and present our contributions.

A. Related Works

Existing work on automated traffic detection spans a spectrum from software-based deep-learning classifiers for encrypted traffic to hardware-accelerated systems that target high throughput. Several studies have demonstrated the effectiveness of end-to-end deep learning models for traffic and encrypted-traffic classification (e.g., convolutional and recurrent architectures, and sequence/CNN-based methods), showing that learned representations can substantially outperform traditional port/flow statistical features in benchmark datasets [14], [15]. At the same time, deploying these models at the network edge introduces some constraints: limited compute, strict energy budgets, and real-time latency requirements, which motivate research into lightweight model designs and on-device inference for IoT/edge contexts. Recent work on lightweight architectures and model compression for edge deployment (e.g., MobileNet-style adaptations and other compact CNNs) shows promising accuracy/efficiency tradeoffs, but most evaluations remain at the algorithmic level or on general-purpose edge CPUs rather than being evaluated on low-power, hardware-accelerated edge platforms [16], [17]. Parallel to these efforts, there is a body of work exploring FPGA/ASIC implementations of packet-level classification and intrusion detection, and a growing systematization of knowledge about FPGA-based network intrusion detection designs that trade off throughput, latency, and flexibility. These FPGA studies make clear that high-performance implementations are possible, yet many are evaluated on server-class or research FPGA boards and focus on throughput rather than the power/area constraints of true edge devices [18].

Two recent papers exemplify the gap we address. A frequency-domain feature extraction pipeline, as proposed in [1], is implemented on a Xilinx Alveo U50 FPGA, with inference performed on the host CPU using CatBoost. While the design achieves high accuracy and multi-Gbps throughput on laboratory datasets, its reliance on a server-class accelerator, offloading of inference to the host, and lack of mechanisms for on-device online adaptation limit its direct applicability to battery- or energy-constrained edge nodes. In contrast, the work in [19] focuses on enhancing Low Power Wide Area (LPWA) IoT security through tamper-resistant Hardware Security Modules (HSMs) and Quantum Key Distribution (QKD). While this approach strengthens key management and physical protection, it does not introduce lightweight, real-time traffic detection primitives suitable for low-power edge deployments. Moreover, it assumes the availability of costly infrastructure, such as HSMs and QKD links, which are impractical to scale across resource-constrained endpoints. Likewise, ZETROS [3] represents a complementary direction that tackles IoT security from a zero-trust perspective rather than pure traffic analysis. It integrates a lightweight PUF-based mutual-authentication handshake with distributed anomaly detection, blacklisting, and smart-contract–driven trust scoring. This design enables continuous verification of nodes and mitigates both insider and outsider threats across heterogeneous IoT networks, but it primarily focuses on protocol- and trust-level protection instead of designing lightweight, on-device traffic classification models. Similarly, PLLM-CS [21] introduces a transformer-based intrusion detection framework for satellite and IoT networks that adapts pre-trained large language models to encode contextual dependencies within multivariate traffic data. By converting network features into tokenized sequences, the model leverages self-attention mechanisms to capture long-range correlations across flows and achieves near-perfect accuracy on UNSW-NB15 and TON-IoT datasets without fine-tuning. While it demonstrates the potential of attention-based models for cyber-threat detection, its reliance on high-capacity transformer backbones and centralized training makes it impractical for constrained, real-time edge environments. Finally, the Multi-Feature Hybrid Anomaly Detection framework [22] proposes an integrated model that fuses Autoencoder (AE), Isolation Forest (IF), XGBoost, Random Forest, and LSTM networks to jointly capture spatial and temporal patterns in Industrial Control System (ICS) data. By combining unsupervised reconstruction, ensemble-based classification, and sequence modeling, it significantly reduces false alarms while maintaining high precision and recall across SWaT and Wind Turbine SCADA datasets (F1-score = 0.98). This multi-feature fusion enables robust and interpretable detection of complex cyber-physical anomalies, yet the design remains primarily evaluated at the algorithmic level rather than on constrained or embedded edge devices.

B. Contributions

Taken together, prior art shows three recurring limitations relevant to real-time, low-power edge detection: (1) many high-accuracy ML approaches are evaluated without realistic low-power hardware targets or with inference performed off-device; (2) FPGA/accelerator implementations emphasize throughput on server-class boards rather than energy-optimized System-on-Chips (SoCs) or small FPGAs suitable for the network edge; and (3) system-level security proposals (e.g., HSM/QKD) improve key resilience but do not solve the core problem of continuously monitoring and classifying traffic on constrained edge nodes.

Our approach addresses these gaps by introducing a fully hardware-integrated, low-power architecture that performs feature extraction and lightweight inference directly on a constrained edge platform. We evaluate latency, throughput, and energy per inference under realistic LPWA and edge traffic

scenarios and compare the design against representative baselines, including host-offloaded FPGA extraction with CatBoost and lightweight software models. By enabling on-device, real-time detection without reliance on server-class accelerators or heavy cryptographic infrastructure, the proposed architecture provides a practical solution for scalable edge security.

III. Proposed Approach

In this section, we propose **LEMD-GARF**, a hardware-efficient approach for real-time malware detection at the edge.

A. System Overview

LEMD-GARF is designed to operate efficiently on resource-constrained edge devices while maintaining high detection accuracy. As illustrated in Fig. 1, the system comprises three primary stages: (1) data preprocessing and feature extraction, (2) intelligent feature selection using a GA, and (3) classification using a RF classifier optimized for hardware deployment. Unlike traditional cloud-centric approaches, our architecture performs all computations locally at the edge, eliminating the need for external communication and enabling low-latency, privacy-preserving detection.

The design philosophy prioritizes computational efficiency without sacrificing accuracy. By reducing the feature space through intelligent selection and employing a lightweight yet powerful classifier, the system achieves real-time performance suitable for deployment on low-power-based processors commonly found in IoT gateways and edge computing nodes. The entire pipeline, from raw packet capture to classification decision, is optimized to minimize both execution time and energy consumption, which are critical factors for battery powered or energy constrained deployments.

B. Dataset and Preprocessing

Our approach utilizes network flow-based features derived from the CIC-IDS dataset family [20], which contains contemporary attack scenarios including DDoS, port scanning, botnet traffic, and various intrusion patterns mixed with benign network activity. Each flow is characterized by statistical features extracted from bidirectional packet sequences, including temporal metrics (inter-arrival times, flow duration), volumetric features (packet counts, byte statistics), and protocol-specific attributes (TCP flags, window sizes). The preprocessing stage performs several critical operations to ensure data quality as follows:

1) **Data Consolidation:** Multiple CSV files containing network flows from different capture sessions are merged into a unified dataset, preserving temporal and statistical integrity.
2) **Cleansing:** Infinite values resulting from division-by-zero operations (common in real-time feature extraction) are identified and replaced with NaN markers. Rows containing missing values are subsequently removed to prevent model instability.
3) **Deduplication:** Duplicate flows, which may arise from multi-interface captures or logging errors, are eliminated to prevent bias during training.
4) **Normalization:** Feature scaling is applied using standard score normalization (z-score) to ensure all features contribute proportionally to the learning process, particularly important for distance-based and gradient-based algorithms.

Following preprocessing, the dataset is partitioned using stratified sampling into training (80%) and testing (20%) subsets, ensuring balanced representation of attack classes in both sets. This stratification is crucial given the often-imbalanced nature of network intrusion datasets where benign traffic typically dominates.

C. Genetic Algorithm-based Feature Selection

Network flow analysis traditionally relies on numerous statistical features, many of which may be redundant, irrelevant, or computationally expensive to extract in real time. To overcome this challenge, we adopt a two-stage feature selection strategy that combines both filter and wrapper methods:

Mutual Information Filtering. We first apply Mutual Information (MI) scoring to quantify the statistical dependency between each feature and the target labels. Features exhibiting MI scores below a threshold $\theta_{MI} = 0.01$ are pruned, as they contribute negligible discriminative information. This filter-based approach rapidly eliminates clearly irrelevant features before more expensive wrapper-based search begins. MI is computed as:

$$MI(X_i; Y) = \sum_{x \in X_i} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{1}$$

where X_i represents feature i and Y denotes the class labels.

Genetic Algorithm Optimization. On the filtered feature subset, we deploy a GA to search for the optimal feature combination that maximizes classification performance while minimizing cardinality. The GA operates as follows:

1) **Encoding:** Each candidate solution is represented as a binary chromosome where bit i indicates whether feature i is included (1) or excluded (0).
2) **Fitness Function:** Solutions are evaluated using 2-fold stratified cross-validation with a Decision Tree classifier as the base estimator, computing weighted F1-score as the fitness metric. This choice balances computational cost (2-fold vs. 5-fold) with reliable performance estimation during the evolutionary search.
3) **Population:** We maintain a population of 50 individuals, providing sufficient diversity for exploration while remaining computationally tractable.
4) **Genetic Operators:**
   - *Crossover* (probability = 0.5): Uniform crossover exchanges feature selections between parent solutions.
   - *Mutation* (probability = 0.2): Bit-flip mutation randomly toggles feature inclusion/exclusion.
   - *Selection*: Tournament selection (size=3) identifies promising candidates for reproduction.

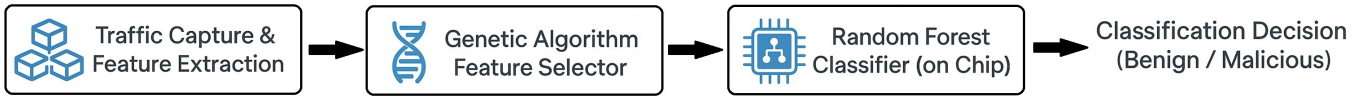

Fig. 1. **LEMD-GARF** Overview

5) **Termination:** Evolution continues for a maximum of 30 generations or until no improvement is observed for 10 consecutive generations (early stopping).

The GA consistently converges to feature subsets of approximately 25 features, a dramatic reduction from the original 80+ features in the raw dataset. This reduction translates directly to lower computational cost during inference, as fewer features must be extracted and processed for each incoming flow.

D. Random Forest Classification

The selected feature subset feeds into a RF classifier, chosen for its strong performance on tabular data, inherent parallelism, and suitability for hardware acceleration. RF is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of their individual predictions.

Architecture. Our RF configuration employs 100 decision trees, each trained on a bootstrap sample of the training data with feature bagging applied at each split. The bootstrap aggregating (bagging) process reduces variance and improves generalization, while the random feature selection at nodes ($\sqrt{n_{features}} \approx 5$ features per split) decorrelates the trees and enhances ensemble diversity.

Training. Each tree is grown using the Gini impurity criterion for split quality:

$$Gini(S) = 1 - \sum_{c=1}^{C} p_c^2 \tag{2}$$

where p_c is the proportion of samples belonging to class c in node S. Trees are grown to maximum depth with minimum samples per leaf set to 1, allowing the forest to capture fine-grained decision boundaries. Overfitting is mitigated by the ensemble averaging effect.

Inference. For a test sample $\mathbf{x}$, each tree T_j produces a class prediction $\hat{y}_j = T_j(\mathbf{x})$. The final prediction is determined by majority voting:

$$\hat{y}_{RF}(\mathbf{x}) = \text{mode}\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{100}\} \tag{3}$$

Alternatively, class probabilities can be obtained by averaging the proportion of votes across the forest, enabling confidence-based filtering or risk-adaptive responses.

E. Hardware Implementation

We implement a hardware verification prototype of **LEMD-GARF** for malicious traffic detection at the network edge by mapping the algorithm directly to an FPGA. The algorithmic structure (ensemble of decision trees) and the numerical operations (node comparisons and leaf-value accumulation) are synthesized in VHDL and are organized around a pipelined datapath: feature inputs are preprocessed and stream into parallel Tree Evaluator modules, each of which uses pipelined multiply-add processing elements to evaluate split functions and compute partial scores. Model parameters (split thresholds and leaf scores) are stored in on-chip BRAM and accessed by the evaluators under the control of a lightweight Finite State Machine (FSM) that schedules tree traversal and memory reads. Partial results from the evaluators combine in a Vote/Average Aggregator to produce the final classification score. The entire design targets the Xilinx XCU200-FSGD2104-2-E device and is functionally verified using the Vivado simulator; this hardware-centric verification confirms bit-accurate correspondence between the VHDL implementation and the reference software model while demonstrating a high-throughput, low-latency inference pipeline. Fig. 2 shows the modular design of the RF-based classifier pipeline.

F. Model Explainability and Trust

For interpretability, we extract global and local explanations:

Global Importance. Feature importance scores are computed by aggregating the total reduction in Gini impurity attributed to each feature across all trees in the forest. This reveals which flow characteristics (e.g., packet rates, IAT statistics, flag counts) are most influential for distinguishing malicious from benign traffic.

Local Explanations. For individual predictions, we trace the decision path through the forest to identify which features and thresholds triggered the classification. This provides actionable insights, such as "This flow was classified as malicious primarily due to abnormally high SYN flag count (23) and short inter-arrival time variance (0.003 ms)."

These explainability mechanisms enhance trust in the automated system and facilitate debugging, model refinement, and compliance with regulatory requirements in safety-critical deployments.

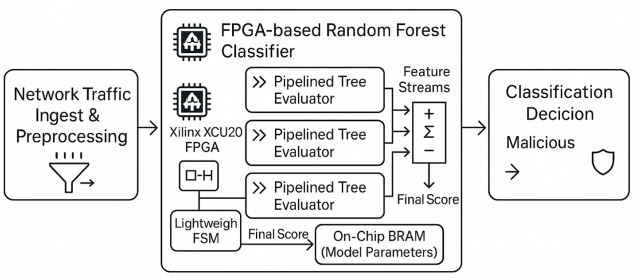

Fig. 2. Architecture of Proposed Hardware Accelerator

TABLE I
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ZETROS (Zero-Trust Framework) [3]	99.83	99.70	99.75	99.72
PLLM-CS (LLM-based Framework) [21]	95-100	94-99	94-99	94-99
Hybrid ML for ICS [22]	99.10	98.80	98.70	98.75
FPGA-based Frequency Transformation [1]	97.72	94.43	99.18	96.75
LEMD-GARF (Proposed Method)	**98.19**	**99.56**	**98.19**	**98.83**

IV. RESULTS AND ANALYSIS

In this section, we discuss the experimental setup, simulations, and evaluation results.

A. Overall Performance Metrics

Table I presents a comparison of **LEMD-GARF** with state-of-the-art methods in IoT security and intrusion detection. The comparison includes recent approaches ranging from zero-trust frameworks to federated learning and large language model-based solutions. As shown, **LEMD-GARF** achieves competitive performance with an accuracy of 98.19%, precision of 99.56%, recall of 98.19%, and F1-score of 98.83%. While some recent methods achieve slightly higher accuracy (e.g., ZETROS at 99.83%), the proposed system offers significant advantages in computational efficiency, hardware feasibility, and real-time processing capability, making it more suitable for resource-constrained edge environments.

The high precision score shows that flagged malicious traffic is almost always correct, while balanced precision and recall indicate effective detection of both common and rare attacks.

B. Per-Class Classification Analysis

Fig. 3 shows the normalized confusion matrix of **LEMD-GARF**, highlighting per-class classification performance. Most attack categories are identified with high precision, while misclassifications mainly occur among web-based attacks.

LEMD-GARF demonstrates perfect classification accuracy for several attack types that represent common threats in IoT and edge computing environments. Specifically, it achieves 100% detection accuracy for bot traffic, DDoS attacks, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, FTP-Patator, Infiltration, PortScan, and SSH-Patator, and 99% accuracy for DoS Slowloris, highlighting its strong capability in identifying a wide range of malicious behaviors.

Web-based attacks present classification challenges. "Web Attack - Brute Force" achieves 53% accuracy, with 40% of instances misclassified as "Web Attack - XSS", while "Web Attack - XSS" achieves 78% accuracy, with 21% misclassified as "Web Attack - Brute Force". In addition, "Web Attack - SQL Injection" achieves 75% accuracy, with 25% misclassified as "DoS GoldenEye". These results indicate that different types of web attacks exhibit similar network flow characteristics, leading to inter-class confusion.

C. Computational Efficiency Analysis

LEMD-GARF demonstrates excellent computational efficiency suitable for edge deployment. The inference latency of 0.189 µs per sample enables real-time processing of network traffic flows, supporting high-throughput scenarios commonly encountered in edge computing environments. This performance is achieved through the optimized feature selection process, which reduced the original feature space from over 80 features to approximately 25 features without significant accuracy degradation. The system's efficiency is further enhanced by the lightweight RF implementation, which requires minimal computational resources compared to deep learning alternatives while maintaining competitive accuracy. The ensemble nature of RF provides inherent parallelism opportunities that can be exploited in hardware implementations.

D. Hardware Implementation Results

The FPGA implementation of **LEMD-GARF** is synthesized and verified on the Xilinx XCU200-FSGD2104-2-E device. The design achieves a stable clock frequency of 143.22 MHz, processing each feature vector in 27 clock cycles, which corresponds to an average inference latency of 0.189 µs per sample. This throughput (5.29 MSamples/s) confirms that the architecture fully meets the real-time processing requirements for edge-level malicious traffic detection. Functional verification using the Vivado simulator demonstrates bit-accurate equivalence between the VHDL implementation and the software

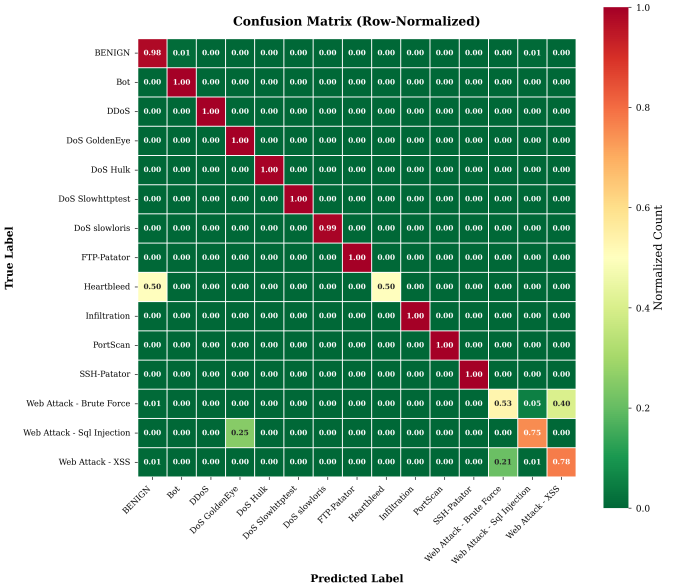

Fig. 3. Normalized Confusion Matrix Illustrating the Per-Class Classification Performance of **LEMD-GARF**

TABLE II
FPGA RESOURCE UTILIZATION AND POWER CONSUMPTION COMPARISON

Method	LUT	LUTRAM	FF	BRAM	DSP	IO	BUFG	Static Power (W)
LEMD-GARF (Proposed Method)	516	150	1483	0.5	19	81	1	2.476
FPGA-based Frequency Transformation [1]	88000	NR	NR	103.5	12	NR	NR	10.827

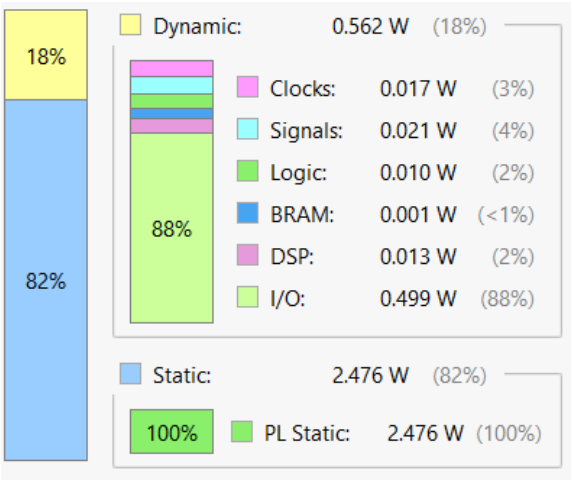

Fig. 4. Power Consumption of Proposed Hardware Accelerator

LEMD-GARF model. The pipelined datapath and parallel tree-evaluator enables low-latency, high-throughput classification, while a lightweight FSM manages BRAM access and tree traversal without introducing timing bottlenecks.

Table II summarizes the FPGA resource utilization and power consumption of the proposed hardware implementation compared to FPGA-based Frequency Transformation approach [1]. The results demonstrate that the proposed architecture achieves significantly lower resource usage and total power consumption, highlighting its suitability for low-power, resource-constrained edge environments.

The measured dynamic and static power consumption post-implementation is shown in Fig. 4. The proposed FPGA implementation of **LEMD-GARF** demonstrates reasonable power characteristics due to its fully on-chip processing pipeline and lightweight control logic. By eliminating external memory access and PCIe data transfers, the design reduces dynamic switching activity, resulting in lower total on-chip power compared to prior FPGA-assisted frameworks [1]. This enables reliable, low-cost real-time edge deployment by minimizing energy per inference.

V. CONCLUSION

This paper proposed **LEMD-GARF**, a hardware-efficient malware traffic detection framework for resource-constrained edge environments. By combining GA-based feature selection with an RF classifier, the framework reduces feature dimensionality and on-device processing overhead while maintaining high detection accuracy. FPGA implementation demonstrates real-time, low-power operation with significantly lower resource usage than existing FPGA-based approaches, confirming its suitability for edge deployments. Future work will explore adaptive learning and heterogeneous hardware platforms.

REFERENCES

[1] Z. Hu, H. I. Hasegawa, Y. Yamaguchi, and H. Shimada, "Enhancing detection of malicious traffic through FPGA-based frequency transformation and machine learning," *IEEE Access*, vol. 12, pp. 2648–2659, 2024.

[2] O. Jaiswal, P. Asare, J. Gadgilwar, K. Rahangdale, P. Adekar, and L. Bitla, "Malicious address identifier (MAI): A browser extension to identify malicious URLs," in *Proceedings of the 11th International Conference on Emerging Trends in Engineering and Technology–Signal and Information Processing (ICETET-SIP)*, pp. 1–5, 2023.

[3] C. A. Baykara, I. Safak, and K. Kalkan, "ZETROS: A zero-trust IoT network security framework using distributed blacklisting, trust scoring and smart contracts," *Computer Networks*, vol. 271, article 111601, 2025.

[4] A. Rayes and S. Salam, *Internet of Things from Hype to Reality*. Springer, 2019.

[5] G. Santucci, "The internet of things: Between the revolution of the internet and the metamorphosis of objects," *Visions and Challenges for Realising the Internet of Things*, vol. 1, pp. 11–24, 2010.

[6] G. Fortino, F. Messina, D. Rosaci, and G. M. Sarnè, "ResIoT: An IoT social framework resilient to malicious activities," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1263–1278, 2020.

[7] M. Nezarat, E. Khedersolh, H. Sh. Shahhoseini, and A. Rezaei, "ML-based real-time URL inspection with hardware acceleration for enhanced web security," in *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*, pp. 1–6, 2025.

[8] M. Gelgi, Y. Guan, S. Arunachala, M. S. S. Rao, and N. Dragoni, "Systematic literature review of IoT botnet DDoS attacks and evaluation of detection techniques," *Sensors*, vol. 24, no. 11, article 3571, 2024.

[9] A. Affinito, S. Zinno, G. Stanco, A. Botta, and G. Ventre, "The evolution of Mirai botnet scans over a six-year period," *Journal of Information Security and Applications*, vol. 79, article 103629, 2023.

[10] H. Liu, F. Han, and Y. Zhang, "Malicious traffic detection for cloud-edge-end networks: A deep learning approach," *Computer Communications*, vol. 215, pp. 150–156, 2024.

[11] A. Mudgerikar, P. Sharma, and E. Bertino, "Edge-based intrusion detection for IoT devices," *ACM Transactions on Management Information Systems*, vol. 11, no. 4, Dec. 2020.

[12] M. Wang, B. Zhang, X. Zang, K. Wang, and X. Ma, "Malicious traffic classification via edge intelligence in IIoT," *Mathematics*, vol. 11, no. 18, article 3951, 2023.

[13] T. Bhattacharya, A. V. Peddi, S. Ponaganti, and S. T. Veeramalla, "A survey on various security protocols of edge computing," *The Journal of Supercomputing*, vol. 81, article 310, published Dec. 2024.

[14] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, May 2019.

[15] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, pp. 445–458, 2019.

[16] C. Sun, B. Chen, Y. Bu, S. Zhang, D. Zhang, and B. Jiang, "Lightweight traffic classification model based on deep learning," *Wireless Communications and Mobile Computing*, vol. 1, article 3539919, 2022.

[17] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, "Energy consumption of on-device machine learning models for IoT intrusion detection," *Internet of Things*, vol. 21, article 100670, 2023.

[18] L. Le Jeune, A. Sateesan, M. M. Rabbani, T. Goedeme, J. Vliegen, and N. Mentens, "SoK: Network intrusion detection on FPGA," in *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 242–261, 2021.

[19] H. -S. Han, T. -H. Choi, and J. -S. Yoon, "Enhancing security in low-power wide-area (LPWA) IoT environments: The role of HSM, tamper-proof technology, and quantum cryptography," *Journal of Web Engineering*, vol. 23, pp. 787–800, 2024.

[20] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, pp. 357–374, 2012.

[21] M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, and D. Sharma, "PLLM-CS: : Pre-trained Large Language Model (LLM) for cyber threat detection in satellite networks," *Ad Hoc Networks*, vol. 166, issue C, 2025.

[22] M. M. Aslam, A. Tufail, L. C. De Silva, and R. Apong, "Multi-feature hybrid anomaly detection in ICS: An integration of ML, DL, and statistical techniques," in *ACM Workshop on Secure and Trustworthy Deep Learning Systems (SecTL)*, pp. 43-51, 2025.