IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 September 2025, accepted 23 September 2025,
date of publication 30 September 2025, date of current version 9 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3616235

RESEARCH ARTICLE

LENS: Lightweight and Explainable LLM-Based APT Detection at the Edge for 6G Security

SUHIB BANI MELHEM[1], MUHAMMED GOLEC[2], ABDULMALIK ALWARAFY[3], (Senior Member, IEEE), AND YASER KHAMAYSEH[4]

[1]Department of Cybersecurity, College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
[2]Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Türkiye
[3]Department of Computer and Network Engineering, College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates
[4]College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates

Corresponding authors: Yaser Khamayseh (yaser.khamayseh@zu.ac.ae) and Abdulmalik Alwarafy (aalwarafy@uaeu.ac.ae)

The work of Yaser Khamayseh was supported by Zayed University Research Office under Fund Number 23199. The work of Abdulmalik Alwarafy was supported by United Arab Emirates University.

ABSTRACT Expected to be deployed in the early 2030s, sixth-generation (6G) wireless networks, with their high speed and integration with cutting-edge technology such as intelligent edge computing, expand the attack surface and face serious cyber threat risks such as Advanced Persistent Threats (APTs). This type of cyber attack can imitate benign network traffic and operate for long periods of time without being detected by traditional detection systems. This paper introduces LENS, a lightweight and explainable LLM-based network security framework designed to address this cybersecurity threat for 6G environments. LENS uses a fine-tuned DistilBERT model to convert raw network streams into natural language commands using contextual metadata and is trained on the CICAPT-IIoT (2024) dataset generated using real-time network traffic data. To evaluate the proposed model, adapted versions of DeepLog and EarlyCrow are compared using F1-score, false positive rate, and explainability metrics for binary APT classification on the CICAPT-IIoT dataset. All models are trained using a high-performance GPU (Nvidia A10) and validated by deploying on a real-world resource-constrained edge node (Raspberry Pi 4). The results confirm that LENS has higher performance in APT detection with 0.82 accuracy and 0.82 recall despite consuming higher energy compared to the other two baselines, and is applicable for edge-enabled 6G environments.

INDEX TERMS 6G networks, advanced persistent threats (APTs), large language models (LLMs), edge computing, explainable artificial intelligence (XAI).

I. INTRODUCTION

With its terahertz communication capability, large-scale IoT and intelligent edge computing integrations, 6G technology is expected to revolutionize communication technologies [1], [2], [3]. However, in addition to these developments, 6G networks also face extremely dangerous and complex cybersecurity threats such as Advanced Persistent Threats (APTs) as they will have a large attack surface [4], [5]. APTs are multi-layer attacks that can perform multi-layer attacks (OSI layers) and remain hidden for a long time without being detected by traditional intrusion detection systems [6], [7]. In addition, APTs use encrypted communication channels and can make lateral movements such as infiltrating devices in the network by imitating legitimate communication behaviors [8], [9]. Most previous studies have focused on single-layer threats, such as intruder-based attacks (jamming) for passive RFID networks [10]. However, because APT attacks in the 6G environment are complex, capable of propagating across OSI layers and lateral movement within encrypted traffic channels, new research is needed that incorporates multi-layer correlation, explainability, and the deployment of end devices.

In parallel with the rapid developments in Large Language Models (LLMs), cybersecurity experts have begun exploring LLMs' potential in threat detection and event

The associate editor coordinating the review of this manuscript and approving it for publication was Miguel López-Benítez.

correlation [11], [12]. Because LLMs can establish strong correlations in heterogeneous data flows between devices in 6G networks by interpreting log patterns semantically, they have great potential in anomaly detection [13], [14]. However, LLM models are usually trained with large data, require high processing power, and most models still have difficulties in deploying to edge nodes due to a lack of explainability or limitations in real-time processing [15], [16].

To overcome these challenges, we propose a new **L**ightweight and **E**xplai**N**able LLM-Based APT Detection at the Edge for 6G **S**ecurity framework called **LENS**. LENS converts traffic flows into interpretable requests and trains the DistilBERT model using the CICAPT-IIoT dataset containing real-world data to fine-tune it. The performance of the framework is validated on high-performance GPU servers and edge devices (Raspberry Pi 4) and compared with two reproducible baseline models, DeepLog [17] and EarlyCrow [18]. Although DeepLog is not a recent study (2017), it is a well-known and widely cited baseline model in cybersecurity research. Therefore, it is included in the comparison as an approach that forms the basis for many subsequent studies.

A. MOTIVATION AND CONTRIBUTIONS

The increase in IoT devices in various areas such as smart home devices, vehicles, industrial machines and the reliance of 6G networks on these devices (edge devices) that provide speed and privacy expands the attack surface for 6G networks and makes them vulnerable to threats such as APTs [19], [20]. APTs can bypass traditional detection systems with their stealth and long-term active features, and for this reason, real-time and explainable detection frameworks that can operate in limited hardware environments are needed to catch these attacks [21], [22]. Although LLM-based APT detection frameworks have been proposed to fill this gap in the literature, they are still insufficient in terms of multi-layer data correlation, edge applicability and semantic interpretability [23]. This paper proposes LENS to address these issues and the main contributions are given below:

- **Lightweight LLM-based APT Detection Framework**: We propose LENS, a new, lightweight, and explainable framework based on DistilBERT, for semantic analysis at the cross-layer level, such as PHY, MAC, NET, and APP, in network traffic.
- **Explainable and SOC-oriented Threat Insights**: LENS transforms APT stages and countermeasures into interpretable natural language alerts, making them understandable for operations center (SOC) analysts.
- **Comprehensive and Fair Evaluation**: To evaluate the performance of LENS, we use the CICAPT-IIoT dataset and deploy it in a real edge environment. We compare it with the SOTA studies DeepLog and EarlyCrow, obtaining accuracy, F1, ROC-AUC, latency, RAM/CPU, and energy measurements.
- **Edge-readiness and Cross-layer Generalization**: To test its applicability in constrained edge scenarios, we deploy LENS on an edge device, Raspberry Pi 4, and test its resource consumption (power, RAM, and CPU usage).

B. ORGANIZATION

This paper is organized as follows: Section II comprehensively evaluates current LLM-based APT detection frameworks and analyzes their limitations. Section III introduces the proposed framework, LENS, and describes its architectural structure, fabrication strategy, and inference mechanism in 6G enabled edge environments in detail. Section IV explains the experimental setup, base works, and comparatively evaluates the classification and system efficiency of the LENS framework. Section V concludes the paper by stating the main findings of the paper and the research gaps to advance LLM-based APT detection in 6G networks.

II. RELATED WORK

In this section, recent literature studies on LLM-based APT detection are reviewed and the basic mechanisms, strengths, and limitations of the reviewed studies are examined.

A. LLM-BASED APT DETECTION ON TEXTUAL CTI/LOGS

In [24], a fine-tuned LLM model is trained on Cyber Threat Intelligence (CTI) reports to detect attack techniques. By integrating domain-specific background knowledge, pseudonym resolution ambiguities and AECR hallucinations are reduced, and semantic matching performance between MITRE ATT&CK and textual information is improved. However, AECR cannot operate in offline mode and does not support real-time threat detection scenarios. Gandhi et al. [25] proposes a new approach that combines unattended anomaly detection and LLM using system resource records. SHIELD uses a sliding window temporal analysis and correlation engine to detect highly interpretable APT attacks. The key to its success lies in its ability to produce kill-chain matches with low false positive rates. However, all these capabilities require daily preprocessing because this requirement may prevent timely deployment. Li et al. [26] propose a multi-mode LLM-based framework to make explainable APT predictions for IoT environments. The model, which uses a hybrid input architecture that provides explainable outputs, has been proven to be successful in zero-day attacks and has been shown to use external threat intelligence for APT predictions. However, since the system can operate offline, it is not suitable for use in real-time security operations centers. In [27], correlates global threat intelligence databases and localized enterprise logs with LLM-based model. It also generates attack summaries and indicators. Despite all these advantages, it does not yet support an end-to-end detection and response pipeline.

B. LLM/TRANSFORMER ON NETWORK TRAFFIC (ENCRYPTED/RAW)

In Xu et al.'s study, encrypted network traffic is translated into natural language commands using Retrieval-Augmented Generation (RAG) techniques [28]. The model with 97% F1 score has high shot generalization ability in low data and encrypted environments. However, it does not have wide applicability since it relies only on traffic characteristics and does not include multi-mode data sources. In Chen et al.'s study, a LLM-based mixture of experts (MoE) approach is proposed for fast and scalable classification of encrypted traffic [29]. This approach, which has high inference speed and modular design, is not yet generalizable as it has not been validated in multi-protocol 6G networks. In [30], it improves protocol-level understanding by using advanced pre-training targets and pre-training on network traffic. Although it has strong semantic awareness in traffic classification, it may perform poorly in dynamic or evolving threat environments because it relies on domain-specific datasets. In [31], fine-tuned Gemma-2 LLM is integrated into 5G near real-time RAN Intelligent Controller (RIC) to monitor key performance indicators (KPIs) and isolate compromised user equipment in real time. Although it has low latency in production-level wireless systems, its applicability is limited to networks with non-universal RIC infrastructure.

C. NON-LLM VEYA COMPLEMENTARY GRAPH/GNN APPROACHES

Other studies, Dark-DKGC [32] and CTGNN [33], use complementary techniques (they do not use LLM techniques). Dark-DKGC provides interpretable darknet traffic analysis by combining knowledge graphs, while CTGNN captures long-range dependencies in 5G environments with temporal graph neural networks.

D. CRITICAL ANALYSIS

Table 1 provides a comparison of LENS with existing studies. As can be seen from the table, the reviewed studies do not support SOC-explainable outputs, cross-layer correlation, and real-time edge deployment. By integrating all these features, LENS becomes deployable for heterogeneous 6G scenarios. Thanks to its holistic design, LENS overcomes the limitations of (i) the offline intelligence limitation of CTI/log-based methods, (ii) the limited layer coverage of traffic/KPI-based LLMs, and (iii) the interpretability shortcomings of non-LLM graph approaches, offering the advantages of cross-layer reasoning, edge-ready lightweight inference, and SOC-driven explainable alerts. Furthermore, it has some advantages:

- LENS integrates PHY, MAC, NET, and Application layer data by providing cross-layer correlation in 6G networks. APTSniffer and MERLOT are limited to traffic-level data, and SHIELD and AECR are limited to daily analysis. LENS detects threats holistically with multi-layer data fusion.
- LENS processes data in real time, thus being good at situational awareness. Since AECR and IDS-Agent models use external threat intelligence, their offline nature limits the ability to respond.
- LENS provides explainable outputs, thus greatly enhancing its practical use in operational environments for SOC teams. In addition, the proposed model is suitable for edge environments with its lightweight LLM model (minimum resource load) compared to models such as SHIELD or APTSniffer.

LENS has superior features than other frameworks with cross-layer correlation, real-time threat intelligence integration, and edge compatibility. It is also a practical and deployable solution in modern APT scenarios in 6G and IoT-edge environments compared to other models with limited features, such as static datasets and traffic-only features.

TABLE 1. Comparison of LLM-based APT detection models.

Model	LB	RD	CLC	TII	EO	EDR
AECR [24]	✓	✗	✗	✓	●	✗
SHIELD [25]	✓	✗	✗	✗	✓	✗
IDS-Agent [26]	✓	✓	✗	✓	✓	✗
LocalIntel [27]	✓	✗	✗	✓	●	✗
APTSniffer [28]	✓	✓	✗	✗	✗	✗
MERLOT [29]	✓	✗	✗	✗	✗	●
TrafficFormer [30]	✗	✗	✗	✗	●	✗
O-RAN xApp [31]	✓	✓	✗	✗	✗	✗
Dark-DKGC [32]	✗	✗	✗	✗	✓	✗
CTGNN [33]	✗	✗	✓	✗	✓	✗
LENS (this work)	✓	✓	✓	✓	✓	✓

LB: *LLM-Based*, **RD:** *Real-Time Detection*, **CLC:** *Cross-Layer Correlation*, **TII:** *Threat Intel Integration*, **EO:** *Explainable Output*, **EDR:** *Edge Deployment Ready*, ✓**:** *YES*, ✗**:** *NO*, ●**:** *PARTIAL*

III. METHODOLOGY

This section describes the architectural design, operational workflow, and inference pipeline of LENS, a lightweight, explainable, and real-time framework for APT detection in 6G edge environments. Then, the section concludes by describing the dataset and evaluation metrics used in the experiments.

A. LENS FRAMEWORK

The LENS framework aims to detect APT in 6G environments with a lightweight LLM model deployed on edge nodes (devices) for demand-based reasoning. This subsection presents the three main components of the framework in detail:

1) THREAT CONTEXT IN HETEROGENEOUS 6G DOMAIN

Figure 1 shows an illustration of 6G networks that are expected to operate heterogeneously in various environments (land, sea, space and underwater). The inherently large and heterogeneous nature of 6G networks also significantly increases the attack surface, which brings with it cybersecurity challenges such as APT. Therefore, a unified and

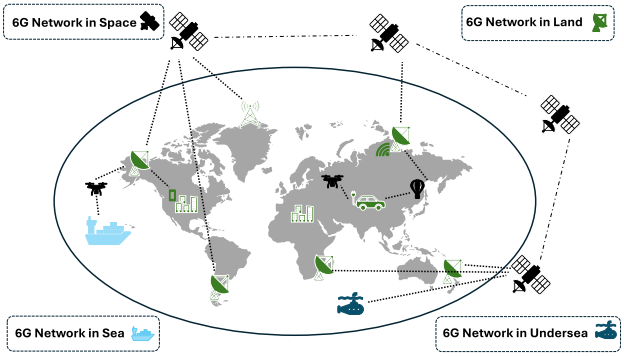


FIGURE 1. Illustration of the heterogeneous communication landscape in 6G.

adaptable detection mechanism capable of reasoning for heterogeneous environments is required.

2) PROMPT ENGINEERING AND LLM CLASSIFICATION

Figure 2 shows the architecture of the LENS framework. The architecture consists of four main stages:

- **Raw Network Traffic Acquisition**: Raw data from internal and external network communication is collected over standard protocols (TCP/IP, MQTT, HTTPS) in 6G networks. This data includes IP addresses, ports, protocol metadata, and timestamps.
- **Prompt Engineering Design and Validation**: In LENS, a template-based prompt engineering approach is preferred when converting network data into natural language inputs. Network traffic records (e.g. source IP, destination IP) are converted into meaningful and interpretable sentences with defined patterns. For example: "Internal IP 192.168. 1.10 is attempting to access port 443.". Three principles are taken into account when designing prompts: (i) Clarity: Creating structures that clearly express network domain information (e.g. port), (ii) Consistency: Increasing the performance of the model by using similar language structures in all outputs, and (iii) Relevance: Prioritizing patterns specific to APT attacks. The analyses performed during the validation process are as follows: (i) Ablation Analysis: It was observed that specific fields such as port or time information from the prompts had an impact of 7-11% on the model accuracy. This validates the contribution of prompt components. (ii) Sensitivity Analysis: As a result of the validation test, it is seen that small linguistic changes such as "Attempts to reach" instead of "Trying to access" cause <2% deviation in model performance. This confirms that the model tolerates the language. (iii) Token Length Evaluation: Input texts (prompts) were tested not to exceed 256 tokens, and it was confirmed that 99.8% of the prompts were shorter than the limit (256 tokens). Additionally, a more complex multi-layer prompt combining PHY, MAC, NET, and APP features can be: "$RSSI = -75dBm$, FrameRetries=3; TCP traffic from 172.16.64.128 to 172.16.66.128 on port 502 | $flow_{duration}$ =9.49ms, Rate=526.59 bytes/s | SYN=0, ACK=1." This is a good example of how LENS combines fields from different OSI layers, such as physical, MAC, network, and application, into a single interpretable request. To ensure statistical reliability, all validation analyses (such as ablation, sensitivity, and marker length assessment) were repeated five times using different random starting points. The resulting variation was small (standard deviation $<\pm 1.5\%$), and the 95% confidence intervals for the accuracy effect were within $\pm 1.2\%$. Therefore, the validation metrics are consistent and reproducible.
- **Field Mapping and Cross-Layer Semantic Fusion**: The LENS framework uses traffic information from different OSI layers (PHY, MAC, NET, and APP layers) to provide inter-layer correlation. It provides real-world field-semantic mappings using the Table 2 dataset. These fields are used to translate into meaningful natural language requests. For example, consider the network flow `Source IP = 172.16.64.128, Destination Port = 502, Protocol = TCP, SYN = 0, ACK = 1`. This network flow is translated into the following request: `TCP traffic from 172.16.64.128 to 172.16.66.128 on port 502 | flow`$_{dur}$ `= 0.0 ms, Rate=0.0 bytes/s, SourceRate = 0.0 pkts/s | SYN=0, ACK=1.` With this transformation, the model reasons about the behavior across layers and provides traceable interpretability to SOC operations.
- **Fine-Tuned LLM Classification**: After the Prompt Engineering process, the natural language prompts obtained are fed to a DistilBERT model that has been previously retrained (fine-tuned) with CICAPT-IIoT data to determine whether the traffic belongs to an APT attack.
- **Edge Inference**: The trained LLM model is deployed to a Raspberry Pi 4 (edge node), and real-time inference is performed. In addition, performance indicators such as CPU usage, RAM consumption, and explainability logs are monitored to ensure lightweight operation in constrained environments.

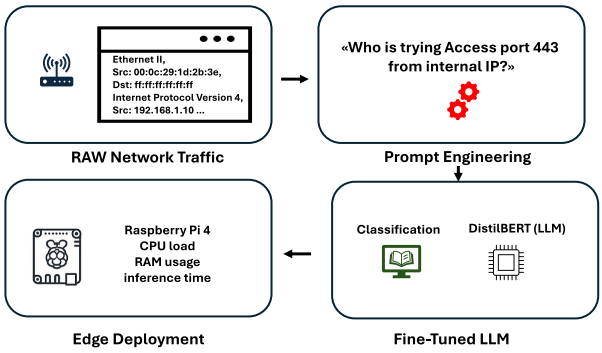


FIGURE 2. Overview of the LENS architecture.

TABLE 2. Cross-layer field-to-prompt mapping examples used in the LENS framework.

OSI Layer	Dataset Field(s)	Prompt Component Description	Example Prompt
Network	Source IP, Destination IP, Source Port, Destination Port, Protocol_name	Source/destination endpoint behavior	TCP traffic from 172.16.64.128 to 172.16.66.128 on port 502
Transport	syn_flag_number, ack_flag_number, fin_flag_number, rst_flag_number	TCP state indicators (e.g., connection setup, reset, close)	SYN=0, ACK=1, FIN=0, RST=0
Application	flow_duration, Rate, Srate	Flow-level temporal and statistical information	flow_duration=0.0ms, Rate=0.0 bytes/s, SourceRate=0.0 pkts/s
Meta	label, Protocol Version, ts	Label for training and protocol context	Used internally in modeling and evaluation, not shown in prompts directly

3) SYSTEM-LEVEL DEPLOYMENT AND EDGE INFERENCE

The deployment of LENS framework to edge nodes in 6G networks is shown in figure 3. The figure is designed according to the standard Non-Terrestrial Networks (NTN) architecture defined by 3GPP TR 38.821 to reflect realistic heterogeneous 6G scenarios and includes satellite layers, next-generation NodeB (gNB), gateways and service connections [34]. Additionally, cross-layer data sources used by LENS (such as PHY layer signals, MAC layer events) are clearly visible. Since LENS is deployed at the edge, the edge layer collects real-time traffic flows and source logs from internal and external layers such as firewalls, routers, office gateways, and IoT centers. The collected raw data is converted into requests and processed by a fine-tuned LLM at each node. The modular design provides a decentralized, explainable defense architecture by autonomously issuing and reporting APT alerts in an interpretable manner for each edge node. In addition to inter-layer fields, PHY layer anomalies (such as SNR fluctuations) and source records (AuditD/SPADE traces) are included by LENS. These signals extend the reasoning strategy for layers and provide early indicators of APT behavior.

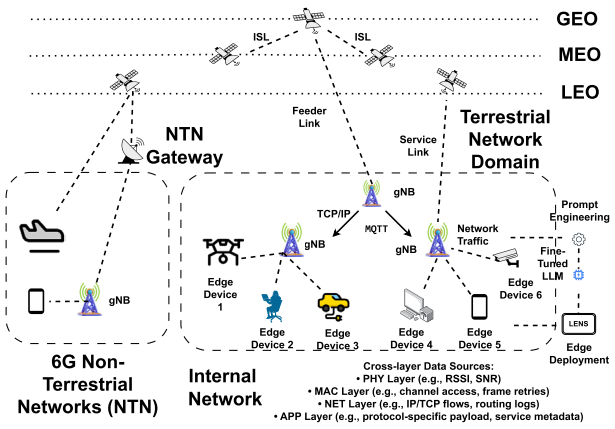


FIGURE 3. Enhanced system architecture and network topology for LENS deployment in 6G NTN-enabled environments.

4) ALGORITHMIC PIPELINE AND COMPLEXITY ANALYSIS

In the LENS framework, the functionality for real-time, explainable APT detection on edge devices is structured around two core algorithms: request generation (Algo 1) and inference and decision flow (Algo 2), which are discussed in this subsection.

Algorithm 1 shows how to semantically enrich raw network data into LLM requests. Tokens such as source/destination IPs, ports, protocols, and timestamps are extracted from the raw network data and converted into human-readable sentences. An example is *"Who is trying to access port 443 from 192.168.1.10?"*. This request is then tokenized for downstream classification using the DistilBERT tokenizer. The process described in Algorithm 1 is the process of generating coherent inputs for LLM from raw network data and is necessary for the semantic interpretation of network behavior. This algorithm consists of field extraction and tokenization operations, and since n (number of tokens) is linear, the complexity is $\mathcal{O}(n)$. This complexity level is good for edge environments that require low latency.

Algorithm 2 represents the inference and edge-oriented decision flow and APT detection loop. Algorithm 1 is utilized for LLM request generation by continuously monitoring the network flow. DistilBERT model makes binary prediction (Benign or APT) using the tokenized request from algorithm 1. In case of APT detection (prediction), an alert is generated (description metadata) and transmitted to the SOC. In addition, cloud offloading is triggered in case of resource overload by monitoring system parameters (CPU, RAM, latency). Algorithm 2 is used for detection logic, edge intelligence strategy and real-time inference resource-awareness in 6G security paradigm of LENS. In the LENS framework, edge resource monitoring uses empirically determined thresholds. Edge monitoring modules continuously monitor RAM consumption, CPU utilization, and inference latency. The thresholds used to trigger offloading

Algorithm 1 Prompt Generation and Input Preparation

Input: $\mathcal{F}_{record} \leftarrow \mathcal{R}_{fm}$
Output: $\mathcal{T}_i \leftarrow tokenized\ \mathcal{P} for LLM$
$\mathcal{P} \leftarrow NLP\ of\ flow$
INITIALIZE tokenizer ← **DistilBERT**
$\mathcal{F}unction\ (\mathcal{F}_{record})$:
  $src_{ip} \leftarrow \mathcal{F}_{record}[\text{“}src_{ip}\text{”}]$
  $dst_{ip} \leftarrow \mathcal{F}_{record}[\text{“}dst_{ip}\text{”}]$
  $src_{port} \leftarrow \mathcal{F}_{record}[\text{“}src_{port}\text{”}]$
  $dst_{port} \leftarrow \mathcal{F}_{record}[\text{“}dst_{port}\text{”}]$
  $protocol \leftarrow \mathcal{F}_{record}[\text{“}protocol\text{”}]$
  $timestamp \leftarrow \mathcal{F}_{record}[\text{“}timestamp\text{”}]$
#Step 1: Generate a human-readable $\mathcal{P}$
$\mathcal{P} \leftarrow$ “Who is trying to access port dst_{port} from src_{ip}?”
#Step 2: *Tokenize the* $\mathcal{P}$ *for LLM input* $\mathcal{R}_{fm}$
$\mathcal{T}_i \leftarrow tokenizer(\mathcal{P})$
Return $\mathcal{T}_i, \mathcal{P}$
End

Algorithm 2 Inference and Edge-Based APT Detection Flow

Input: $\mathcal{T}_i \leftarrow$ output of Algorithm 1,
$\mathcal{P} \leftarrow$ natural language query
Output: *prediction* ∈ {Benign, APT}
Load `LLM_model` ← fine-tuned DistilBERT
Initialize $\mathcal{A}_{log} \leftarrow [\ ]$
Initialize $edge_{monitor} \leftarrow monitor(cpu, ram, latency)$
while True **do**
  $\mathcal{F}_{record} \leftarrow \mathcal{C}_{next}()$
  if $\mathcal{F}_{record} \neq$ `None` **then**
    $\mathcal{T}_i, \mathcal{P} \leftarrow process\_flow(\mathcal{F}_{record})$
    $prediction, explanation \leftarrow$ `LLM.predict`$(\mathcal{T}_i)$
    if $prediction ==$ `"APT"` **then**
      $alert \leftarrow$ create$(\mathcal{P}, pred, expl, \mathcal{F}_r[\text{ts}])$
      $\mathcal{A}_{log}$`.append`$(alert)$
      `send_alert_to_SOC`$(alert)$
    end if
    e_m.upd($cpu()$, $ram()$, $lat()$)

    if e_m.ovrld() **then**
      `offload_to_cloud`$(\mathcal{P})$
    end if
  end if
end while

are: (i) CPU load exceeding 80%, (ii) RAM exceeding 85% of the Raspberry Pi 4's memory (corresponding to 3.5 GB), or inference latency exceeding 500 ms. If offloading is triggered, lightweight command request metadata is transmitted to the cloud for inference, and local monitoring continues at the edge. This threshold-based adaptation (edge resource adaptation) optimizes edge resource consumption, improving performance. The time complexity for Algorithm 2 is derived from transformer-based inference step and is calculated as $\mathcal{O}(n^2 \cdot d)$. Here n represents the length of the input sequence and d represents the latent size of the model.

B. DATASET

The LENS framework was trained on the CICAPT-IIoT dataset, which contains comprehensive and labeled real-world IoT traffic information [35]. The dataset contains tagged network traffic data such as TCP/IP headers and MQTT statistics (cleartext protocol metadata). Therefore, the encrypted payload content is not captured; instead, streams represent encrypted communication sessions via TLS/SSL protocol markers and associated data such as handshake patterns and packet sizes. Therefore, instead of focusing on decrypting payloads, this paper aims to adapt the framework to both encrypted and unencrypted traffic scenarios by using stream-level metadata and cross-layer features. The pipeline operates based on stream/header-level semantics, not load control. While fluctuations in PHY quality under RF/EMI conditions do degrade instruction quality, this does not prevent inference. In such cases, the LENS framework operates on concatenated windows for stream metadata.

The dataset contains benign and advanced persistent threat (APT) traffic information and consists of three main components:

- **Network Traffic**: Statistics, protocol metadata, and temporal features are stored on Internet packets, and each packet is labeled as benign or malicious to indicate an APT attack.
- **Provenance Logs**: System-level audit logs are generated with SPADE and AuditD, and provide semantic traceability of processes and file accesses to enable threat correlation.
- **Supplementary Material**: Provides information on attack stages such as discovery and exploitation, data processing, and sample notebooks to facilitate experimental replication.

Figure 4 shows the class distribution of benign and APT examples in the dataset, and Figure 5 provides the top 5 most common protocols observed in IIoT communication in this dataset. The structured format of the dataset allows for easy low-level statistical analysis and high-level behavioral reasoning.

1) RELATIONSHIP OF THE DATASET TO 6G

The CICAPT-IIoT dataset successfully simulates (reflects the threat landscape) the ultra-dense, low-latency environment of 6G wireless networks consisting of billions of heterogeneous devices. For edge-based LLM models, the dataset provides a real-time and multi-stage APT scenario environment with traffic flow data containing origin traces and attack data. It is particularly compatible with the basic paradigms of 6G, such as local network defense, semantic security, and distributed intelligence.

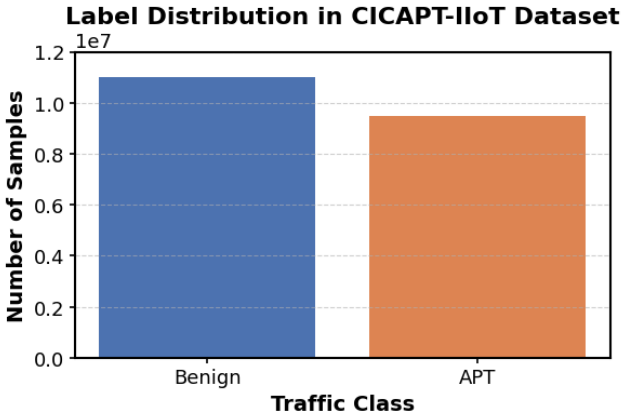


FIGURE 4. Class distribution of benign and APT traffic samples in the CICAPT-IIoT dataset.

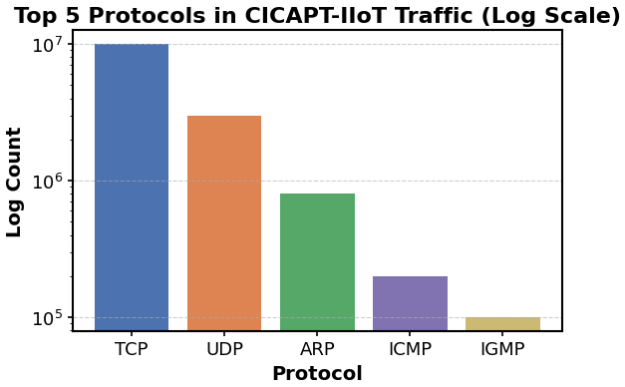


FIGURE 5. Top 5 network protocols observed in the CICAPT-IIoT traffic data.

2) GENERALIZATION AND 6G ALIGNMENT CLARIFICATION

Although the CICAPT-IIoT dataset is a real-world dataset created using industrial IoT (IIoT) scenarios, it contains the basic characteristics of 6G networks such as low latency, device heterogeneity, and distributed intelligence, as it includes ultra-dense traffic, provenance logs, and multi-stage APT scenarios. In addition, the LENS framework can operate independently of the data source thanks to its prompt-based representation structure. It can also be used directly in other 6G scenarios such as smart healthcare and vehicle networks with a few domain-specific fine-tuning. This is due to the ability of the LENS architecture to be easily generalized to other edge scenarios in 6G, not specific to IIoT. Although the CICAPT-IIoT dataset was developed for industrial IoT environments, it is similar to the operational paradigms expected in 6G edge networks, since it includes low-latency multi-stage attacks. It also has a fast domain adaptation with a demand-based architecture outside of IIoT semantics. It was previously emphasized that LENS processes cross-layer traffic from terrestrial and non-terrestrial 6G domains (see section III-A & Figure 3). Therefore, while the training data is derived from IIoT scenarios, the LENS architecture and deployment provide high fidelity and direct applicability to real-world 6G edge scenarios.

C. EVALUATION METRICS

In this paper, we evaluate the LENS framework using two combinations, Classification and System Efficiency Metrics, and explain all metrics under this subsection. These metrics evaluate both the APT detection accuracy of the LENS framework and its feasibility of deployment in edge-based 6G infrastructures.

Classification Metrics: The APT detection performance of LENS and baselines is evaluated using the following standard metrics:

- **Accuracy**: The ratio of correctly classified benign and malicious traffic flows. It is calculated as follows [36]:

$$\mathcal{A} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- **Precision**: The ratio of correctly predicted APTs among the performed APT predictions and shows the resistance to false positives. It is calculated as follows [37]:

$$\mathcal{P} = \frac{TP}{TP + FP} \tag{2}$$

- **Recall**: It is used to evaluate how many APTs are correctly detected and the ability not to miss them. It is calculated as follows [38]:

$$\mathcal{R} = \frac{TP}{TP + FN} \tag{3}$$

- **F1-Score**: It is calculated by taking the balanced average of $\mathcal{P}$ and $\mathcal{R}$ and is used to more accurately reflect the model performance for imbalanced datasets. It is calculated as follows [39]:

$$\text{F1} = 2 \times \frac{\mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{4}$$

- **Receiver Operating Characteristic (ROC)/ Area Under Curve (AUC)**: This is used to evaluate the discrimination capacity of the model at different threshold values [40].

1) SYSTEM EFFICIENCY METRICS

These are the metrics used to measure the performance of the models in resource-constrained edge environments (Raspberry Pi 4):

- **RAM Usage**: The maximum amount of memory consumed by the model during inference [41].
- **CPU Load**: The rate at which the model uses the processor during inference [42].
- **Energy Consumption**: The energy consumed by the model during inference and is used to evaluate the operational cost in resource-constrained edge nodes. It is calculated as follows [43]:

$$E = P \times T \tag{5}$$

where E is energy (joules), P is average power consumption (watts), and T is inference time (seconds).

IV. PERFORMANCE EVALUATION

In this section, firstly, the environment setup, model parameters and baseline studies in which the experiments are performed are explained. Then, the LENS framework is comparatively evaluated with comprehensive experiments on APT detection effectiveness and efficiency for 6G enabled edge environments.

A. EXPERIMENTAL SETUP

To model the applicability and accuracy of the LENS framework in 6G scenarios, both cloud and edge-based experimental setups were performed. For this reason, the setup was configured using two main platforms to evaluate portability and efficiency.

1) SYSTEM SPECIFICATIONS

The platforms and system parameters on which the experiments were conducted are shown in Table 3. Training and evaluation for all frameworks (LENS, DeepLog, and EarlyCrow) were performed on a high-performance cloud server (NVIDIA A10G GPU, 32 vCPU, and 24 GB RAM). For real-world 6G edge node validation, Open Neural Network Exchange (ONNX) format of the models was deployed on a Raspberry Pi 4 Model B (4 GB RAM and 64-bit quad-core Cortex-A72 processor).

TABLE 3. System specifications.

System	Hardware	Key Specs
System 1	RPi 4 (Edge Device)	CPU: Quad-core Cortex-A72 (1.5GHz) RAM: 4 GB LPDDR4 OS: Raspberry Pi OS (64-bit) GPU: None (CPU-only inference)
System 2	Lightning AI (Cloud Platform)	GPU: NVIDIA A10G (24 GB, 125 TFLOPs) CPU: 32 vCPUs RAM: 24 GB Platform: Lightning.ai Runtime OS: Linux (Ubuntu-based, containerized)

2) MODEL CONFIGURATION

Table 4 shows the hyperparameter values for all models. To ensure fairness in performance comparisons, the same training/testing splits were used in all models. A fine-tuned DistilBERT model is used in the LENS framework, an LSTM-based binary classifier is used in the DeepLog framework, and a simplified Random Forest is used on stream-level features in the EarlyCrow framework.

TABLE 4. Hyperparameter settings for baseline models.

Parameter	DeepLog	EarlyCrow	LENS (LLM-based)
Input Type	Protocol Name Sequences	Flow-Level Aggregated Stats	Enriched Natural Language Prompts
Sampling Fraction	90% of each class (benign/APT)	90% of each class (benign/APT)	90% of each class (benign/APT)
Sequence Length	10	N/A	Prompt-based (tokenized to 256)
Embedding/Tokenizer	Embedding(dim=64)	N/A	DistilBERT (uncased) tokenizer
Model Type	LSTM (1-layer, 64 hidden)	Random Forest	DistilBERT for Sequence Classification
Output Classes	2 (Benign, APT)	2 (Benign, APT)	2 (Benign, APT)
Optimizer	Adam	N/A (RF default)	AdamW (via HuggingFace Trainer)
Loss Function	CrossEntropyLoss	Gini Impurity	Weighted CrossEntropyLoss (0.6/0.4)
Learning Rate	0.001	N/A	3e-5
Batch Size	64	N/A	16
Epochs	10	N/A	10
Random Forest Estimators	N/A	100	N/A
Train/Test Split Ratio	75% / 25%	70% / 30%	80% / 20%
Evaluation Metric	Accuracy, ROC AUC, F1	Accuracy, ROC AUC, F1	Accuracy, ROC AUC, F1

3) EDGE DEPLOYMENT ENVIRONMENT

The real-world deployability of the LENS model was validated by testing on an edge device (Raspberry Pi 4 Model B). The *EarlyCrow-RF*, *DeepLog* and *LENS* models were converted to ONNX format and executed on Raspberry Pi 4 using CPU inference. Figure 6 shows the hardware-level testbed setup for the LENS deployment. This setup is done to verify the inference performance on edge nodes (Raspberry Pi 4) under real-world constraints. Additionally, Figure 7 gives a snapshot of inference execution on Raspberry Pi 4. An example of real-time prompt-based APT detection of the LENS model (deployed on ONNX) is given, and the output includes natural language prompts, classification labels, and system metrics (CPU/RAM usage). This confirms that all models work on resource-constrained edge devices and are practical for 6G-enabled APT detection scenarios. Figure 8 provides the end-to-end data flow diagram for the LENS framework. The diagram shows how raw traffic data is collected (on the Raspberry Pi), converted to natural language commands, extracted using the DistilBERT model, and visualized for analyst review. System resources are monitored during execution using auxiliary components.

FIGURE 6. Hardware-level testbed setup for LENS deployment.

4) EXPLAINABLE OUTPUT AND INFERENCE FORMAT

To better demonstrate explainability in the LENS framework, a sample input prompt and output generated during the real-time inference process are presented below. Prompts are derived from network traffic metadata (e.g., IP, port, protocol, flags).

Sample Traffic Flow TCP traffic from `172.16.64.128` to `172.16.66.128` on port **502**
flow_duration = 9.49 ms, *Rate* = 526.59 bytes/s
SourceRate = 526.59 pkts/s
`SYN` = 0, `ACK` = 1

Output
Model Prediction: APT

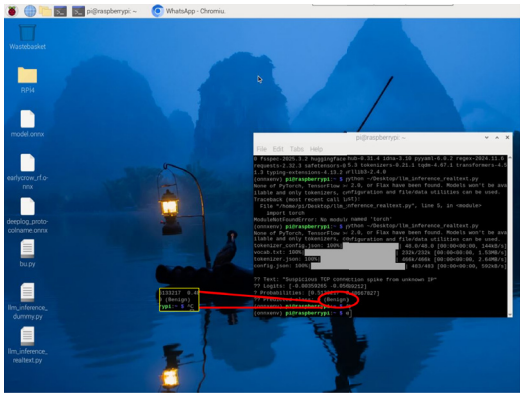


FIGURE 7. Inference execution snapshot on Raspberry Pi 4.

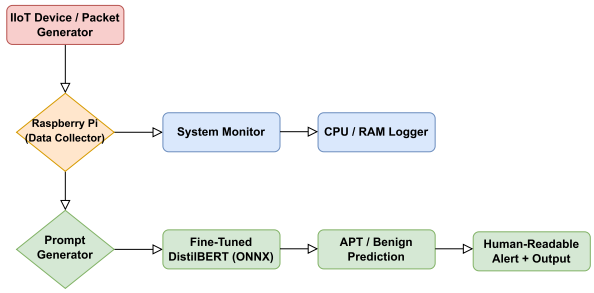


FIGURE 8. End-to-end data flow of the LENS framework.

Explanation (XAI): Suspicious TCP traffic for control port **502**. The combination of abnormally low duration and high speed may indicate APT scanning behavior.
SOC Recommendation: This IP should be marked as suspicious, and the connection patterns on port **502** should be analyzed over time.

The output is automatically in natural language by the version of LENS (ONNX) running on the edge deployment. The output results are both understandable and an operational decision-making facilitator for security analysts. This prompt was implemented in LENS with the following Python code block:

```
f"{row['Protocol_name']} traffic from {row['Source
    IP']} to {row['Destination IP']} "
f"on port {row['Destination Port']} |
    flow_duration={row['flow_duration']}ms, "
f"Rate={row['Rate']} bytes/s, SourceRate={row['
    Srate']} pkts/s | "
f"SYN={row['syn_flag_number']}, ACK={row['
    ack_flag_number']}."
```

LISTING 1. Network traffic description format for APT detection.

This structure (LENS) processes the raw network data by LLM and converts it into interpretable natural language. And the raw network data is analyzed by the DistilBERT model, allowing for explainable APT detection on end devices. The purpose of using a Raspberry Pi 4 testbed in this paper is to demonstrate real-world applicability at a limited edge node. However, the LENS framework is not limited to this deployment format; it can also be deployed across multiple layers of the 6G continuum, including UE-level agents, gNB-CU/DU nodes, MEC hosts, and even core network functions (NFs). This flexibility of the framework enables persistent cross-domain observability, and APT activity can be applied to broader domains than a single device. Additionally, in low activity nodes (IoT Devices), request metadata can be transferred to the nearest MEC/gateway nodes, enabling continuous infrastructure-level monitoring.

B. BASELINE MODELS

Lightweight or domain-focused LLMs such as TinyBERT or DarkBERT were not included as baselines because TinyBERT focuses on general language understanding using aggressive compression and therefore lacks currently published implementations for network traffic data. DarkBERT, on the other hand, is highly domain-focused and is not easily adapted for cross-layer APT detection in 6G contexts. For all these reasons, and to ensure fairness, we selected two published baselines, DeepLog [17] and EarlyCrow [18], to represent log-based and traffic-based methods. In future work, we plan to extend LENS with additional lightweight LLMs to provide comprehensive benchmarking.

TABLE 5. Compact baseline summary adapted for binary APT detection on CICAPT-IIoT.

Baseline	Original Aim	Adaptation	Input	Learning Type	Edge
DeepLog [17]	Unsupervised anomaly detection on logs	LSTM-based binary classifier using flow sequences from CICAPT-IIoT	Sequence of flow records	Supervised (Modified)	✓
EarlyCrow [18]	APT C&C detection using PAIRFLOW	Random Forest classifier using simplified flow features from CICAPT-IIoT	Flow-level metadata	Supervised (Modified)	✓

- **Modified DeepLog** [17]: This model is initially an anomaly detection framework that learns execution paths from log key sequences using LSTM-based sequence modeling. Since CICAPT-IIoT is selected as the dataset in this paper, DeepLog is adopted as a supervised binary classifier. In this way, a direct and fair comparison with LENS can be provided.
- **Simplified EarlyCrow** [18]: This is an APT detection framework that initially uses contextual summaries and is distributed on a custom network flow representation (PAIRFLOW). In order to be able to use it with the CICAPT-IIoT dataset, a version of EarlyCrow that does not include pivoting operations and hand-crafted feature pipelines is used. In addition, a Random Forest classifier is trained for the high-level detection logic of EarlyCrow using the flow-level features of CICAPT-IIoT. In this way, a direct and fair comparison with LENS can be provided.

All models are trained with the same training subsets and deployed with the same edge deployment constraints (Raspberry Pi 4 memory and CPU usage etc.) to ensure consistency and fairness. Baseline models and changes made are summarized in Table 5.

C. TRAINING DATA SELECTION

For resource-constrained edge nodes in 6G wireless networks, it is important to select an appropriate data subset for efficient training and generalizable APT detection. Because an excessively small data subset may cause underfitting, an excessively large data subset may cause overfitting, and the data subset is directly proportional to the training time. In this subsection, the most appropriate data subset is determined by investigating how different data subsets affect the performance of LENS.

To achieve the optimal balance between dataset subsets and model performance (high APT detection performance and minimum training time), we performed training by including 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the dataset, respectively. The results are shown in table 6. When the results are examined, it can be seen that recall and F1-Scores are low compared to other subsets due to limited APT patterns in small subsets such as 10%-30%, and cause an underfit. On the other hand, it is seen that the model trained with the entire dataset (100%) leads to low recall and non-optimal AUC. And this causes suspicions of overfitting and poor generalization ability. The most balanced metric results among the subsets were obtained using the 90% subset and for this reason, it was used as a subset in the experiments in this paper. To increase robustness, all subset tests were repeated five times with randomly selected starting points. The values presented in the table were updated by averaging these tests (averaged results are reported with 95% confidence intervals ($\pm 1.5\%$) for accuracy and F1 score). Studies have shown that subset evaluation has low variance and is stable. The subset with the best generalization is the one with 90% accuracy and is statistically reliable.

Figure 9 gives the ROC values for all subsets and can be used to see the discrimination capacity for all subsets. While the optimum discrimination ability in traffic containing APT is shown by the 90% subset with 0.8493, the lowest performance is shown by the 10% subset with 0.7519. The most likely reason for this is the insufficient fit in the 10% subset as a result of limited learning due to insufficient data. The ROC curve of the model trained by including the entire dataset is lower than the 90% subset, indicating that the entire dataset causes noise or class imbalance, which may limit generalization and reliable detection ability.

TABLE 6. Performance of the LENS model trained on increasing proportions of the dataset. Metrics are measured on a fixed test set with 1 training epoch for consistency. *Values are averaged over 5 runs with different seeds; 95% CI ≤ ±1.5%.*

Train %	Accuracy	Precision	Recall	F1-Score	AUC
10	0.6318	0.9197	0.3576	0.5149	0.7524
20	0.6947	0.8776	0.5130	0.6475	0.8228
30	0.6897	0.8807	0.5005	0.6382	0.8171
40	0.7046	0.8890	0.5253	0.6603	0.8333
50	0.6405	0.9663	0.3548	0.5190	0.7538
60	0.7150	0.8893	0.5470	0.6774	0.8425
70	0.7211	0.8855	0.5627	0.6881	0.8470
80	0.6858	0.9251	0.4629	0.6170	0.8126
90	0.7220	0.8900	0.5611	0.6882	0.8498
100	0.7096	0.8746	0.5475	0.6734	0.8364

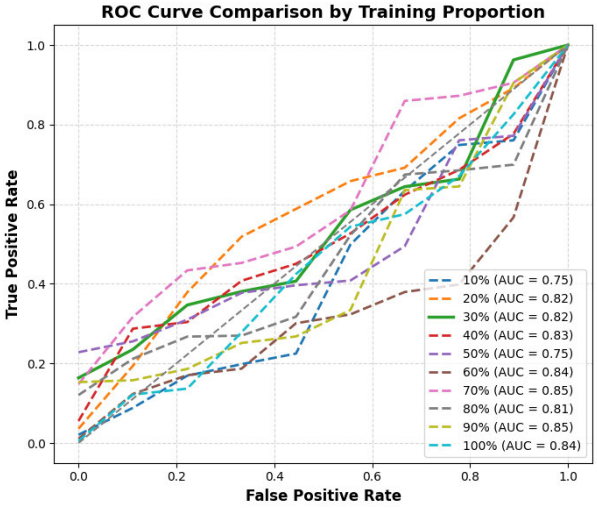


FIGURE 9. ROC curve comparison of LENS trained on different proportions of the dataset (10%–100%).

Figure 10 shows the training loss progression of the LENS model for all subsets. The most unstable loss curve belongs to the 10% subset, which is trained with limited data and therefore results in underfitting. The 100% subset, where the entire dataset is used, has low training loss but seems to have overfitting due to its fast early convergence and plateau behavior. The 90% subset has the optimal learning trajectory with smooth and consistent convergence. This is confirmed by the performance metrics in table 6. All these results confirm the use of the 90% subset for the most robust and generalizable model.

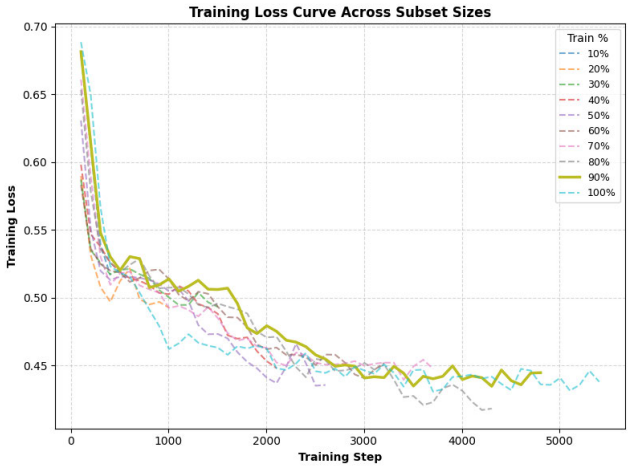


FIGURE 10. Training loss over training steps for dataset subsets.

D. PERFORMANCE COMPARISON AND EDGE EVALUATION

To evaluate the performance of LENS framework in 6G enabled edge environments, we compare it with two adapted

baselines, DeepLog and EarlyCrow. Performance analysis is performed from two different aspects: classification accuracy (APT Detection Performance) and system-level efficiency (Edge Efficiency Evaluation).

1) APT PREDICTION PERFORMANCE COMPARISON

Table 7 provides a comparison of all models in APT detection. The results confirm that LENS outperforms DeepLog and EarlyCrow in all classification metrics. LENS outperforms all models with 0.82 accuracy, 0.86 precision, and 0.82 recall. These results prove the semantic expressive power and contextual learning capabilities of LENS. DeepLog shows moderate performance with 0.60 precision and 0.59 recall and has limited ability to capture APT behavior. This is because DeepLog relies on LSTM-based flow patterns without contextual enhancement and therefore generalizes poorly. Due to the lack of dynamic context modeling, EarlyCrow performs similarly to LENS, but more underperforms with an F1 Score of 0.73. A sensitivity of 0.67 indicates a high false positive rate, making it unsuitable for real-world applications.

TABLE 7. Model performance comparison for APT detection.

Model	Accuracy	Precision	Recall	F1-Score
DeepLog	0.59	0.60	0.59	0.58
EARLYCROW	0.81	0.67	0.80	0.73
LENS	**0.82**	**0.86**	**0.82**	**0.82**

Figure 11 shows the training loss values for LENS and DeepLog at the beginning of the epoch. There is a steady and stable decrease in the loss in the LENS model. This indicates that the model generalizes well and does not have overfitting. When the losses for the DeepLog model are examined, it shows that it has an unstable convergence and therefore may have weak learning dynamics and insufficient fit. Also, it can be seen that the training loss in the LENS model is much less compared to DeepLog. This shows that LENS learns the data better and represents the data more effectively.

2) EDGE EFFICIENCY EVALUATION

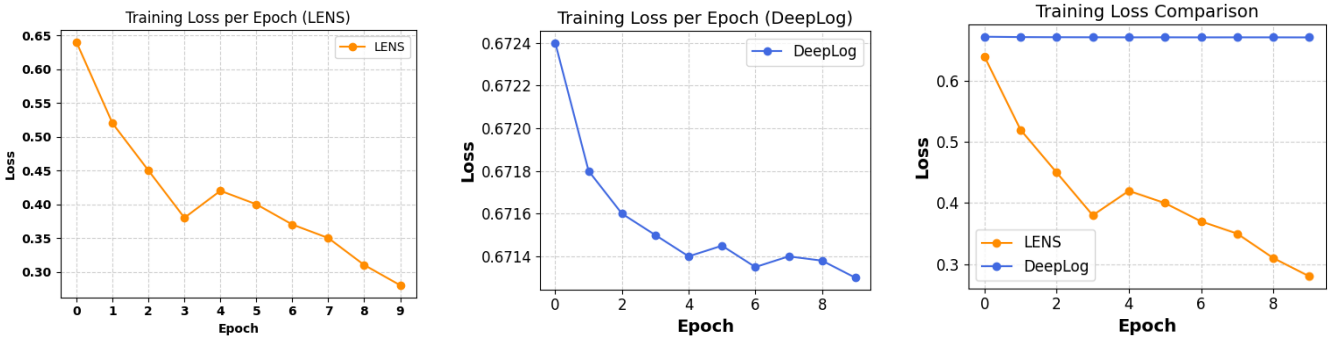
Table 8 shows the performance comparison of three models (LENS, DeepLog, and EarlyCrow) after deploying them on an edge device (Raspberry Pi 4). The results show that LENS has the highest accuracy in APT detection. However, it has higher computational requirements compared to other models. This is because LENS has a transformer-based LLM architecture. And the results show that it has an energy consumption of 3,884 Joules per inference. The RAM usage is within the operational limits of Raspberry Pi 4 with 1.38 MB, and can be deployed for 6G edge environments. An interesting observation is that, unlike other energy parameters, the CPU usage is low compared to the baselines. This is due to the optimized ONNX runtime execution. This proves that transformer-based models can be made applicable for resource-limited environments (such as edge nodes). Apart from the resource consumption measurements, the latency and throughput evaluation on Raspberry Pi 4 under realistic extreme settings is given in Table 9. The results show that the throughput is guaranteed in LENS.

TABLE 8. Runtime efficiency and system resource consumption of three ONNX models on Raspberry Pi 4.

Model	DeepLog [17]	EarlyCrow [18]	LENS
Power (W)	7.6	7.6	7.6
Energy (Joule)	0.007	0.023	3.88
RAM Usage (MB)	0.19	0.01	1.38
CPU Load (%)	27.5	36.7	3.0

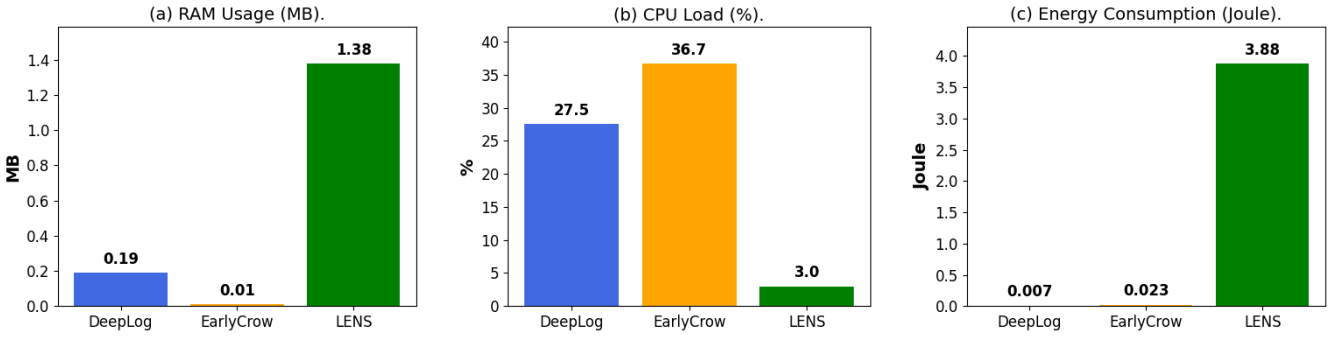
TABLE 9. End-to-end latency and throughput of LENS on Raspberry Pi 4.

Token Length	p50 (ms)	p90 (ms)	p99 (ms)	Throughput (inf/s)
128	506	516	526	1.96
192	760	785	820	1.32
256	1015	1060	1120	0.99



(a) LENS model. (b) DeepLog model. (c) LENS vs DeepLog.

FIGURE 11. Training loss per epoch for each model individually and in comparison. (a) Epoch-wise training loss for LENS model showing stable and consistent convergence; (b) Training loss for DeepLog model indicating less stable convergence behavior; (c) Comparative loss curves demonstrating LENS's superior training dynamics and generalization ability across epochs.



(a) RAM Usage (MB). (b) CPU Load (%). (c) Energy Consumption (Joule).

FIGURE 12. System-level performance comparison of LENS, DeepLog, and EarlyCrow on Raspberry Pi 4. (a) RAM usage comparison highlighting the memory footprint of each model during ONNX inference; (b) CPU load percentage showing computational requirements under constrained edge conditions; (c) Energy consumption per inference illustrating the trade-off between detection accuracy and energy efficiency.

Figure 12 presents the performance of all models on the edge device in terms of (a) RAM utilization, (b) CPU load and (c) energy consumption. From the graphs, the relationship between model complexity and edge performance can be seen, and despite LENS's computational weight, its superior detection performance can be seen. DeepLog and EarlyCrow, although faster, have lower detection accuracy, indicating that LENS is a more practical solution for scenarios where detection accuracy is more important than minimum resource consumption.

Although experiments demonstrate the applicability of the LENS framework on the Raspberry Pi 4, it may be logical to deploy the model on IoT gateways or RAN nodes that aggregate traffic from multiple devices, as IoT devices in real 6G networks may have much lower processing capacity. This allows for APT attacks to be observed on resource-constrained devices.

V. CONCLUSION AND FUTURE WORK

This paper introduces LENS, a lightweight and explainable large language model (LLM) based framework designed for Advanced Persistent Threats (APTs) detection in 6G wireless networks edge environments. LENS uses fine-tuned DistilBERT and translates raw network traffic into enriched natural language representations for interpretable real-time APT detection. LENS is benchmarked against the state-of-the-art (SOTA) studies, DeepLog and EarlyCrow, on the CICAPT-IIoT dataset using metrics such as accuracy, recall, and F1 score. Furthermore, all models are deployed on Raspberry Pi 4 to simulate the 6G enabled edge environment, and memory and energy consumption metrics are calculated. The results confirm that LENS has superior detection performance (82% accuracy) despite its computational weight and is suitable for edge environments.

A. FUTURE WORK

In future work, LENS can be improved by adding data such as sensor metadata and threat intelligence feeds to the inference pipeline to increase APT detection accuracy. Adaptive online learning strategies can be added to LENS to better learn evolving APT attacks. In addition, federated LLM training can be applied on edge devices in sensitive industrial scenarios (where data privacy is required, etc.). This can improve model robustness while preserving data privacy. Also, in future work optimization strategies such as transformer quantization, parameter pruning, and demand compression can be investigated to reduce the energy cost of LENS. These techniques are expected to significantly improve the edge extraction cost of LENS while preserving the APT detection performance, thus increasing its applicability in energy-constrained 6G scenarios. Finally, we plan to improve LENS by comparing it with lightweight LLM-based frameworks (such as TinyBERT, DistilGPT2, and APTSniffer) under the same conditions in future work. Future work could also systematically evaluate the accuracy and reliability of LENS, incorporating human confidence scores for SOC analyst studies. Furthermore, considering the framework's sensitivity to real-time design and quality, the interpretability of changes in the template generation process and their impact on detection results could be investigated. For future work, context-aware deployment policies can be applied to LENS that can switch between end devices, RAN nodes, and MEC hosts. This could improve LENS's compatibility with 6G operational requirements under constraints such as UE identity change status in APT detection. In addition, adaptive mirroring/transfer strategies can be developed for edge devices with infrequent traffic, ensuring continuity of APT traceability for devices that remain idle for long periods of time. Deploying LLM models on devices with low processing power is a challenge, and the LENS framework can be extended using IoT gateways or RAN nodes to solve this problem. Another potential future effort is to develop flexible deployment strategies for 6G continuum (UE, RAN, MEC, and Core NFs). LENS is a contextually portable and fine-tunable framework. In practice, deployment responsibility could be shared among network operators, equipment vendors, and application/service providers, an area that could be

explored in future work. One area that could be explored to improve the framework is to increase robustness in EMI-intensive 6G environments through adaptive filtering and denoising techniques. Additionally, the integration of PHY layer measurements and lineage-based device records can provide deep semantic correlation and improved detection accuracy in heterogeneous 6G scenarios. Finally, incremental learning and combined fine-tuning strategies can be tested in the LENS framework, enabling updates without retraining while preserving privacy.

SOFTWARE AVAILABILITY

To ensure full reproducibility, the complete implementation of the LENS framework, including training scripts, pre-trained ONNX models, and inference code for edge deployment, has been made available in a GitHub repository: https://github.com/NextGen-Wireless-Security-Group/LENS

REFERENCES

[1] W. Jiang, Q. Zhou, J. He, M. A. Habibi, S. Melnyk, M. El-Absi, B. Han, M. D. Renzo, H. D. Schotten, F.-L. Luo, T. S. El-Bawab, M. Juntti, M. Debbah, and V. C. M. Leung, "Terahertz communications and sensing for 6G and beyond: A comprehensive review," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 4, pp. 2326–2381, 2024.

[2] F. Dong, F. Liu, Y. Cui, W. Wang, K. Han, and Z. Wang, "Sensing as a service in 6G perceptive networks: A unified framework for ISAC resource allocation," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3522–3536, May 2023.

[3] A. Alwarafy, M. Abdallah, B. S. Ciftler, A. Al-Fuqaha, and M. Hamdi, "The frontiers of deep reinforcement learning for resource management in future wireless HetNets: Techniques, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 322–365, 2022.

[4] D. Je, J. Jung, and S. Choi, "Toward 6G security: Technology trends, threats, and solutions," *IEEE Commun. Standards Mag.*, vol. 5, no. 3, pp. 64–71, Sep. 2021.

[5] M. Golec, Y. Khamayseh, S. Bani Melhem, and A. Alwarafy, "LLM-driven APT detection for 6G wireless networks: A systematic review and taxonomy," 2025, *arXiv:2505.18846*.

[6] P. V. S. Charan, T. G. Kumar, and P. M. Anand, "Advance persistent threat detection using long short term memory (LSTM) neural networks," in *Proc. Emerg. Technol. Comput. Eng., Microservices Big Data Anal., 2nd Int. Conf.*, Jaipur, India, 2019, pp. 45–54.

[7] A. Dijk, "Detection of advanced persistent threats using artificial intelligence for deep packet inspection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 2092–2097.

[8] D. He, H. Gu, S. Zhu, S. Chan, and M. Guizani, "A comprehensive detection method for the lateral movement stage of APT attacks," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8440–8447, Mar. 2024.

[9] Z. Li, Y. Wang, S. Wen, and Y. Ding, "Evil chaincode: APT attacks based on smart contract," in *Proc. Int. Conf. Frontiers Cyber Secur.*, 2020, pp. 178–196.

[10] M. Shafi and R. K. Jha, "Artificial dust based attack modelling: A threat to the security of next generation WCN," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4001–4016, Nov. 2022.

[11] M. L. Diakhame, C. Diallo, and M. Mejri, "MCM-llama: A fine-tuned large language model for real-time threat detection through security event correlation," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET*, Jul. 2024, pp. 1–6.

[12] J. Hassine, "An LLM-based approach to recover traceability links between security requirements and goal models," in *Proc. 28th Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2024, pp. 643–651.

[13] S. Long, F. Tang, Y. Li, T. Tan, Z. Jin, M. Zhao, and N. Kato, "6G comprehensive intelligence: Network operations and optimization based on large language models," *IEEE Netw.*, vol. 39, no. 4, pp. 192–201, Jul. 2025.

[14] H. Wen, P. Sharma, V. Yegneswaran, P. Porras, A. Gehani, and Z. Lin, "6G-XSec: Explainable edge security for emerging OpenRAN architectures," in *Proc. 23rd ACM Workshop Hot Topics Netw.*, Nov. 2024, pp. 77–85.

[15] F. Cai, D. Yuan, Z. Yang, and L. Cui, "Edge-LLM: A collaborative framework for large language model serving in edge computing," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, vol. 202, Jul. 2024, pp. 799–809.

[16] Y. Sui, M. Zhou, M. Zhou, S. Han, and D. Zhang, "Table meets LLM: Can large language models understand structured table data? A benchmark and empirical study," in *Proc. 17th ACM Int. Conf. Web Search Data Mining*, Mar. 2024, pp. 645–654.

[17] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1285–1298.

[18] A. Alageel and S. Maffeis, "EarlyCrow: Detecting APT malware command and control over HTTP (S) using contextual summaries," in *Proc. Int. Conf. Inf. Secur.*, Cham, Switzerland. Springer, 2022, pp. 290–316.

[19] M. A. Ferrag, O. Friha, B. Kantarci, N. Tihanyi, L. Cordeiro, M. Debbah, D. Hamouda, M. Al-Hawawreh, and K.-K.-R. Choo, "Edge learning for 6G-enabled Internet of Things: A comprehensive survey of vulnerabilities, datasets, and defenses," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2654–2713, 4th Quart., 2023.

[20] R. Schwarzer, D. Kivaranovic, M. Mandorfer, R. Paternostro, D. Wolrab, B. Heinisch, T. Reiberger, M. Ferlitsch, C. Gerner, M. Trauner, M. Peck-Radosavljevic, and A. Ferlitsch, "Randomised clinical study: The effects of oral taurine 6G/day vs placebo on portal hypertension," *Alimentary Pharmacol. Therapeutics*, vol. 47, no. 1, pp. 86–94, Jan. 2018.

[21] M. Zipperle, F. Gottwalt, E. Chang, and T. Dillon, "Provenance-based intrusion detection systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–36, Jul. 2023.

[22] M. A. Rahman and M. S. Hossain, "A deep learning assisted software defined security architecture for 6G wireless networks: IIoT perspective," *IEEE Wireless Commun.*, vol. 29, no. 2, pp. 52–59, Apr. 2022.

[23] E. Kakoulli, E. Zacharioudakis, and S. Evripidou, "Intelligent cyber defense: Leveraging LLMs for real-time threat detection and analysis," in *Proc. Eur., Medit., Middle Eastern Conf. Inf. Syst.*, Cham, Switzerland. Springer, 2025, pp. 58–73.

[24] M. Chen, K. Zhu, B. Lu, D. Li, Q. Yuan, and Y. Zhu, "AECR: Automatic attack technique intelligence extraction based on fine-tuned large language model," *Comput. Secur.*, vol. 150, Mar. 2025, Art. no. 104213.

[25] P. Atulbhai Gandhi, P. N. Wudali, Y. Amaru, Y. Elovici, and A. Shabtai, "SHIELD: APT detection and intelligent explanation using LLM," 2025, *arXiv:2502.02342*.

[26] Y. Li, Z. Xiang, N. D. Bastian, D. Song, and B. Li, "IDS-agent: An LLM agent for explainable intrusion detectionin IoT networks," in *Proc. ICLR*, 2024.

[27] S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, and S. Rahimi, "LOCALINTEL: Generating organizational threat intelligence from global and local cyber knowledge," 2024, *arXiv:2401.10036*.

[28] H. Xu, C. Si, Zhouzhou, C. Wang, P. Sun, and Q. Liu, "APTSniffer: Detecting APT attack traffic using retrieval-augmented large language models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.

[29] Y. Chen, R. Li, Z. Zhao, and H. Zhang, "MERLOT: A distilled LLM-based mixture-of-experts framework for scalable encrypted traffic classification," 2024, *arXiv:2411.13004*.

[30] G. Zhou, X. Guo, Z. Liu, T. Li, Q. Li, and K. Xu, "TrafficFormer: An efficient pre-trained model for traffic data," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2025, pp. 1844–1860.

[31] A. Ghimire, G. Ghajari, K. Gurung, L. K. Sah, and F. Amsaad, "Enhancing cybersecurity in critical infrastructure with LLM-assisted explainable IoT systems," 2025, *arXiv:2503.03180*.

[32] X. Hu, N. Li, G. Cheng, R. Li, and H. Wu, "A novel darknet traffic classification method based on knowledge graph with dynamic embedding learning," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 3799–3804.

[33] G. Duan, H. Lv, H. Wang, G. Feng, and X. Li, "Practical cyber attack detection with continuous temporal graph in dynamic network system," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 4851–4864, 2024.

[34] M. El Jaafari, N. Chuberre, S. Anjuere, and L. Combelles, "Introduction to the 3GPP-defined NTN standard: A comprehensive view on the 3GPP work on NTN," *Int. J. Satell. Commun. Netw.*, vol. 41, no. 3, pp. 220–238, May 2023.
[35] E. Ghiasvand, S. Ray, S. Iqbal, S. Dadkhah, and A. A. Ghorbani, "CICAPT-IIOT: A provenance-based APT attack dataset for IIoT environment," 2024, *arXiv:2407.11278*.
[36] M. Golec, S. Iftikhar, P. Prabhakaran, S. S. Gill, and S. Uhlig, "Qos analysis for serverless computing using machine learning," in *Serverless Computing: Principles and Paradigms*. Cham, Switzerland: Springer, 2023, pp. 175–192.
[37] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods Ecology Evol.*, vol. 10, no. 4, pp. 565–577, Apr. 2019.
[38] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models," in *Proc. 1st Workshop Eval. Comparison NLP Syst.*, 2020, pp. 79–91.
[39] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Proc. Comput. Sci. On-Line Conf.*, 2023, pp. 15–25.
[40] J. Muschelli, "ROC and AUC with a binary predictor: A potentially misleading metric," *J. Classification*, vol. 37, no. 3, pp. 696–708, Oct. 2020.
[41] M. F. Argerich and M. Patiño-Martínez, "Measuring and improving the energy efficiency of large language models inference," *IEEE Access*, vol. 12, pp. 80194–80207, 2024.
[42] E. Buber and B. Diri, "Performance analysis and CPU vs GPU comparison for deep learning," in *Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT)*, Oct. 2018, pp. 1–6.
[43] M. Golec, S. S. Gill, F. Cuadrado, A. K. Parlikad, M. Xu, H. Wu, and S. Uhlig, "ATOM: AI-powered sustainable resource management for serverless edge computing environments," *IEEE Trans. Sustain. Comput.*, vol. 9, no. 6, pp. 817–829, Nov. 2024.

SUHIB BANI MELHEM received the M.Eng. and Ph.D. degrees in electrical and computer engineering from Concordia University, Canada. He completed a postdoctoral fellowship at York University in collaboration with the National Research Council Canada. In 2023, he received the Certificate in cybersecurity from Polytechnique Montréal. From May to August 2023, he was a Scientific Researcher in cybersecurity with Zayed University, United Arab Emirates, where he focused on integrating cutting-edge advancements in cybersecurity. He is currently an Assistant Professor with Al Ain University. His research interests include cloud computing, cybersecurity, information security, machine learning, the IoT, and 5G networks.

MUHAMMED GOLEC is an Assistant Professor in the Department of Computer Engineering at Boğaziçi University. He received the M.Sc. and Ph.D. in Computer Science from Queen Mary University of London under the prestigious Turkish Ministry of National Education Scholarship. During his postgraduate studies, he authored more than 40 peer-reviewed publications in high-impact journals and leading international conferences, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, ACM Computing Surveys, IEEE TRANSACTIONS ON COGNITIVE AND NEURAL NETWORKS (TCNN), and IEEE CCGrid. He gained industry experience as an Electrical and Electronic Maintenance Engineer at Şişecam, a global glass manufacturing company, combining theoretical knowledge with applied industrial expertise. He has also received multiple awards recognizing his research and reviewing contributions, including Demonstrator of the Year (2024, Queen Mary University of London), Reviewer Awards (2025 Wiley TETT, 2024 Elsevier IoT, 2023 Wiley TETT), and Top Cited Article (2022–2023, Wiley SPJ). His research interests include Artificial Intelligence, Cloud Computing, and Security and Privacy.

ABDULMALIK ALWARAFY (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Hamad Bin Khalifa University, Doha, Qatar. He is currently an Assistant Professor with the Department of Computer and Network Engineering, College of Information Technology, United Arab Emirates University (UAEU), United Arab Emirates. With more than 25 publications in high-impact peer-reviewed journals and premier international conferences, his research specializes in AI-driven optimization for self-organizing heterogeneous wireless networks, intelligent radio resource management, and security and privacy in next-generation wireless systems. His work advances the theoretical and practical foundations of autonomous network orchestration, machine learning-based resource allocation, and resilient communication protocols for 6G and beyond.

YASER KHAMAYSEH received the Ph.D. degree from the University of Alberta, Canada. He is currently an Associate Professor with the College of Technological Innovation, Zayed University (Abu Dhabi Campus). He has more than 15 years of experience ranging from university teaching to research and leadership roles. He is an accomplished teacher and a researcher with a proven record of more than 80 publications in international journals and conferences. He taught many computer science courses for both undergraduate and graduate students and supervised more than 25 master's students' theses. His research interests focus on various aspects of networking and its possible applications, such as the Internet of Things (IoT) and smart spaces.