

# A Lightweight Language-Model-Driven Agentic Framework for Intrusion Prediction, Detection, and Mitigation in 6G-Enabled IoT Networks

ALAEDDINE DIAF<sup>1</sup>, ABDELAZIZ AMARA KORBA<sup>1,2</sup> (Senior Member, IEEE),  
WAEEL JAAFAR<sup>3</sup> (Senior Member, IEEE), NOUR ELISLEM KARABADJI<sup>4</sup>, AND  
YACINE GHAMRI-DOUDANE<sup>5</sup> (Senior Member, IEEE)

<sup>1</sup>Laboratoire Réseaux et Systèmes (LRS), Badji Mokhtar - Annaba University, Annaba 23000, Algeria

<sup>2</sup>Computer Science Department, German University of Technology in Oman, Halban Campus, Muscat 130, Oman

<sup>3</sup>École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC H3C 1K3, Canada

<sup>4</sup>National Higher School of Technology and Engineering, Laboratoire De Technologies Des Systèmes Énergétiques (LTSE) E3360100, Annaba 23005, Algeria

<sup>5</sup>L3i, La Rochelle University, 17042 La Rochelle, France

CORRESPONDING AUTHOR: W. JAAFAR (wael.jaafar@etsmtl.ca)

This work was supported in part by the Mitacs Globalink Research Award (GRA) Program and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery under Grant CRSNG RGPIN-2023-03757.

**ABSTRACT** The extensive deployment of Internet of Things (IoT) devices in emerging 6G-enabled environments, characterized by massive connectivity, distributed edge intelligence, and ultra-low-latency communication, allows cyber threats to evade detection until malicious activities manifest. Conventional intrusion detection systems remain fundamentally reactive, identifying threats only after suspicious patterns become observable in network traffic, which hinders both proactive threat anticipation and timely mitigation. To address these limitations, this paper proposes a role-based multi-agent cybersecurity framework for 6G-enabled IoT networks that enables proactive intrusion prediction, real-time detection, and knowledge-driven threat mitigation. The framework establishes a coordinated defense loop composed of three specialized agents: (i) a prediction agent that leverages Small Language Models (SLMs) to model packet-level temporal dependencies and proactively forecast malicious traffic sequences, (ii) a detection agent that performs real-time traffic classification using a lightweight gradient boosting model suitable for low-latency edge environments, and (iii) a knowledge graph-driven mitigation agent that maps detected threats to corresponding defensive countermeasures. Extensive evaluation on the CICIOT2023 dataset demonstrates the effectiveness of the proposed framework, achieving 98.24% accuracy in proactive packet-level detection using forecasted packets and 99.94% accuracy in real-time flow-based detection. The proposed framework combines structured multi-agent coordination, low-latency inference, and knowledge-driven response, making it a promising step toward scalable and intelligent cybersecurity management in 6G-enabled IoT networks.

**INDEX TERMS** Security, agentic AI, attack mitigation, autonomous cyber defense, intrusion prediction, LLMs, SLMs, IoT.

## I. INTRODUCTION

The Internet of Things (IoT) has experienced exponential growth, from consumer applications to critical sectors such as healthcare, defense, and vehicular networks. As IoT evolves toward 6G-enabled networks, massive connectivity, edge intelligence, and ultra-low latency further accelerate this transformation. However, they also broaden the attack surface and shrink the response window [1]. Securing such dynamic environments, therefore, demands cybersecurity mechanisms

that are accurate, efficient, adaptive, and capable of coordinated real-time response.

Recent reports indicate more than 820,000 daily IoT attack attempts in 2025, along with a 46% increase in malware year-over-year [2]. Massive botnets, such as Aisuru, have launched record-breaking 29.7 Tbps Distributed Denial-of-Service (DDoS) attacks, while insecure endpoints contribute to 40% of global DDoS traffic [3], [4]. These challenges are further amplified in dynamic environments such as vehicular

networks and emerging 6G-enabled IoT networks, where rapid topological changes and resource constraints render conventional reactive defenses insufficient [5].

Intrusion Detection Systems (IDS) are widely deployed to monitor network activity and identify potential threats in IoT environments. However, most IDS operate reactively at the flow level [6], [7], detecting attacks only after traffic aggregation introduces latency. This delay is problematic for fast-evolving threats such as ransomware and advanced persistent threats (APTs) [8]. Packet-level approaches provide fine-grained visibility and enable earlier detection of attack indicators, but face scalability challenges in large-scale deployments [9]. In contrast, flow-level methods improve scalability at the expense of delayed detection [10]. Moreover, most existing IDSs lack integrated mitigation capabilities and remain challenged by resource-constrained IoT environments [11], [12], leaving dynamic networks such as the Internet-of-Vehicles (IoV) particularly vulnerable [13].

The transition toward 6G-enabled IoT networks further amplifies cybersecurity challenges due to ultra-low latency requirements, massive device density, highly dynamic network topologies, and distributed edge computing infrastructures. In such environments, conventional reactive intrusion detection mechanisms become increasingly insufficient, as delayed responses may allow attacks to rapidly propagate across interconnected devices and edge nodes. Consequently, 6G-enabled IoT networks require proactive and autonomous cybersecurity frameworks capable of continuously monitoring traffic behavior, anticipating threats before complete attack manifestation, and supporting low-latency adaptive response mechanisms [14].

To address these limitations, recent research [15], [16], [17] has moved toward proactive security approaches that seek to anticipate cyber threats before their occurrence. Although promising, most existing solutions remain limited in scope and primarily focus on isolated prediction tasks, without supporting coordinated decision-making or integrated response mechanisms. In highly dynamic IoT environments, effective defense requires intelligent systems that can continuously monitor network behavior, anticipate emerging threats, and adapt their decisions according to evolving traffic patterns [18].

While recent proactive IDS approaches have leveraged machine learning (ML) to anticipate attacks [15], [19], they often struggle to capture long-range temporal dependencies and subtle contextual patterns in network traffic. Alternatively, Transformer-based Large Language Models (LLMs), which leverage self-attention mechanisms to process sequential data, offer a promising approach for modeling complex temporal dependencies in network traffic. Recent studies [16], [17], [20], [21], [22] have explored their use for proactive intrusion prediction, demonstrating the ability to detect early indicators of malicious activity. However, the high computational and memory requirements of full-scale LLMs limit their applicability in IoT environments.

Small Language Models (SLMs) address this challenge by retaining much of the predictive capability of LLMs while being lightweight and resource-efficient [23], [24]. Their compact design enables frequent model updates, low-latency inference, and scalable operation across distributed and resource-constrained networks. Importantly, SLMs provide an efficient foundation for role-specialized cybersecurity systems capable of continuously monitoring network behavior, anticipating threats, and supporting low-latency decision-making. These capabilities support emerging role-based multi-agent cybersecurity paradigms that leverage structured coordination and knowledge-guided decision-making to enable proactive and context-aware defense in complex IoT networks [25], [26].

In line with the discussion above, we propose a role-based multi-agent framework for proactive cyber defense and cooperative response in IoT networks. The proposed framework integrates prediction, detection, and mitigation within a unified and coordinated network architecture. It leverages SLMs for packet-level intrusion forecasting, a LightGBM-based classifier for real-time detection, and a D3FEND-aligned knowledge graph for knowledge-driven threat mitigation recommendation. The main contributions can be summarized as follows:

- 1) We design a coordinated multi-agent framework in which prediction, detection, and mitigation agents operate collaboratively, ensuring continuous monitoring, adaptive decision-making, and proactive protection. Specifically:
- 2) We propose a proactive intrusion forecasting mechanism that leverages SLMs to predict malicious activity at the packet level, before a network flow is fully completed.
- 3) We develop a real-time detection system based on a resource-efficient LightGBM classifier. It performs traffic classification on the fly, offering low latency and scalability for IoT and edge environments.
- 4) We introduce a knowledge-driven mitigation component that exploits semantic knowledge from the D3FEND<sup>TM</sup> framework. The system maps detected threats to appropriate countermeasures, supporting structured mitigation recommendation generation.
- 5) We validate the efficacy of our approach on the large-scale CICIOT2023 dataset. Experimental results demonstrate early threat detection, high classification accuracy, and practical feasibility in real-world IoT environments, compared to benchmarks.

The remainder of this paper is organized as follows. Section II overviews the related work, while Section III describes the proposed framework. Section IV presents the experimental setup and datasets, and Section V discusses the results along with a benchmarking against state-of-the-art methods. Finally, Section VI concludes the paper.

## II. RELATED WORK

This section reviews recent advances in intrusion detection, prediction, and mitigation in IoT networks, with a focus on AI-driven approaches and their limitations in enabling proactive and coordinated cybersecurity.

IDSs are necessary to secure IoT networks by monitoring traffic and identifying malicious activities. However, the heterogeneous, large-scale, and dynamic nature of IoT environments poses significant challenges for traditional IDS, particularly in achieving real-time detection, adapting to evolving threats, and providing integrated mitigation capabilities [27]. The latter have evolved from signature-based detection toward anomaly detection using classical ML algorithms such as Decision Trees, Random Forests, K-Nearest Neighbors, and Logistic Regression [28]. Ensemble models, such as LightGBM, further demonstrated high detection accuracy with low computational overhead in IoT networks [29]. Although they enable efficient traffic classification, these methods' ability to capture complex and evolving attack patterns remains limited.

Deep learning (DL) models have been increasingly adopted since then to automatically learn discriminative representations from network traffic. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM), capture spatial and temporal patterns, enabling more effective IoT intrusion detection [28], [30], [31]. In parallel, recent work showed that combining ML with feature engineering enhances intrusion detection, in terms of accuracy and false positive events, when using resource-constrained IoT protocols such as MQTT [32]. More recently, hybrid DL architectures and Transformer-based models have been explored to model complex relationships in network traffic using attention mechanisms [33]. However, these approaches often require large training datasets and substantial computational resources, limiting their applicability in real-world 6G-enabled IoT networks. Also, most existing IDS remain inherently reactive, detecting malicious activity only after sufficient attack patterns have manifested in network traffic [34]. Alternatively, generative AI has been explored for cybersecurity in 6G-enabled IoT networks, demonstrating promising intrusion detection capabilities for evolving threat scenarios [35]. Similarly, combined DL with optimization algorithms achieved high detection accuracy while supporting efficient threat detection [36].

Within the last few years, LLM-based hybrid architectures have been proposed. For instance, GPT-BERT-LSTM frameworks [16], LLM agents [37], and BARTPredict [17] have been proposed to forecast traffic for early attack detection. Given their computation overhead, SLMs are seen as an interesting alternative for deployment at the edge of the network. For example, lightweight BERT variants (DistilBERT, MobileBERT, and TinyBERT) enabled efficient attack detection inference with reduced resource requirements [38], [39], [40]. Moreover, SLMs are increasingly recognized as a promising component of agentic AI [25], highlighting

their potential role in coordinated and knowledge-driven cybersecurity systems.

While predictive models provide foresight into potential attacks, effective cybersecurity in 6G-enabled IoT environments requires coordinated mechanisms for real-time response and mitigation support. In fact, traditional mitigation strategies, based on rule-based firewalls and static segmentation, struggle to adapt to evolving threats and the dynamic nature of IoT networks [41]. In parallel, reinforcement learning (RL) has been explored for adaptive mitigation, where agents learn optimal defense policies through simulated interactions [42], [43]. Multi-agent RL (MARL) frameworks have further enabled coordinated distributed responses across IoT nodes, demonstrating resilience against various attack scenarios [44], [45], [46]. Nevertheless, these approaches were computationally intensive and often exhibited limited efficiency–performance trade-offs.

Existing studies mainly focused on isolated aspects of cybersecurity in 6G-enabled IoT networks, such as traffic prediction, intrusion detection, or knowledge-driven mitigation, often treating these tasks independently. However, recent studies emphasized that proactive threat identification, adaptive mitigation strategies, and self-sustaining network operation are key requirements for future end-to-end 6G network automation [14]. In this context, the proposed framework integrates packet-level traffic prediction, flow-level intrusion detection, and mitigation recommendation generation within a coordinated multi-agent architecture. Moreover, unlike computationally intensive RL- and MARL-based defense approaches [47], [48], our framework leverages SLM-driven prediction and knowledge-guided mitigation mechanisms to enable proactive cybersecurity while maintaining computational efficiency for large-scale IoT environments.

## III. PROPOSED FRAMEWORK

This section presents our multi-agent framework, which comprises three complementary autonomous agents that collaboratively perform packet forecasting, intrusion detection, and knowledge-driven mitigation. These agents interact through an agent coordination layer responsible for task orchestration, information exchange, and contextual preservation across the prediction–detection–mitigation cycle. Fig. 1 illustrates the overall architecture and inter-agent data flows.

### A. FRAMEWORK OVERVIEW AND DATA REPRESENTATION

The proposed architecture is structured around three specialized agents: a Prediction Agent, a Detection Agent, and a Mitigation Agent. Through the coordination layer, they operate as a unified system that enables continuous decision flow and adaptive feedback. As illustrated in Fig. 1, data flows sequentially from the Prediction Agent to the Detection Agent and then to the Mitigation Agent, with feedback loops supporting coordinated decision propagation across the framework. Table 1 complements this architectural view by

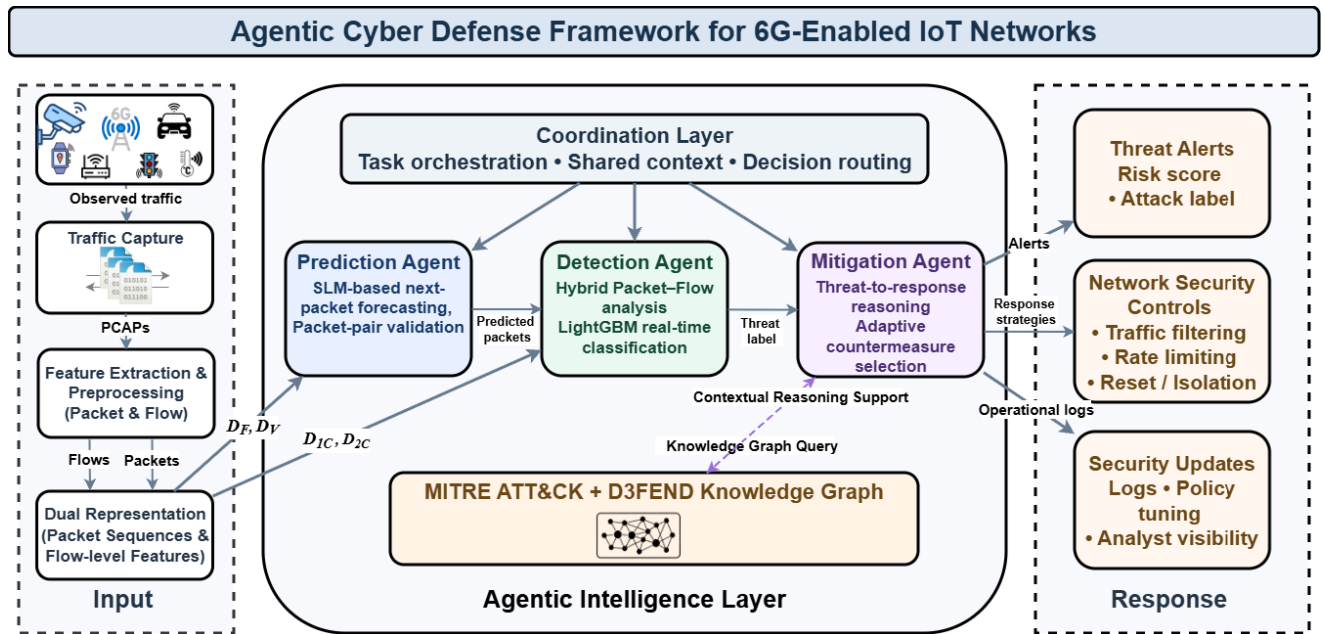


**FIGURE 1.** Pipeline of the proposed multi-agent framework. The Prediction Agent forecasts future packets using SLMs, the Detection Agent combines packet-level and flow-level analysis for proactive intrusion detection, and the Mitigation Agent maps detected attacks to D3FEND-aligned countermeasures through coordinated agent interaction.

**TABLE 1.** Summary of Agent Roles, Inputs, Outputs, and Key Tasks in the Proposed Framework.

| Agent              | Input                                       | Output                                                         | Primary Role                                                   | Key Tasks                                                                                                                                                                                                                                                   |
|--------------------|---------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prediction Agent   | Packet history<br>$S_t = (P_1, \dots, P_t)$ | Forecasted packet<br>$\hat{P}_{t+1}$ , validity score<br>$v_t$ | Next-packet forecasting                                        | (1) Forecast the next packet; (2) Validate forecast consistency; (3) Filter unreliable forecasts based on a confidence threshold.                                                                                                                           |
| Detection Agent    | $\hat{P}_{t+1}, F_t, v_t$                   | $\hat{y}_{\text{packet}, t+1}, \hat{y}_{\text{flow}, t}, a_t$  | Proactive and real-time intrusion detection and classification | (1) Perform flow-level intrusion detection and attack classification; (2) Perform packet-level intrusion detection and attack classification using hybrid packet-flow features; (3) Fuse packet-level and flow-level decisions to generate security alerts. |
| Mitigation Agent   | Alert $a_t$                                 | Mitigation set $\mathcal{R}_t$                                 | Knowledge-driven defense recommendation                        | (1) Map detected attacks to MITRE ATT&CK techniques ( $\mathcal{M}_1$ ); (2) Retrieve D3FEND-aligned countermeasures ( $\mathcal{M}_2$ ); (3) Generate mitigation recommendations.                                                                          |
| Coordination Layer | Outputs from all agents                     | Final alert and mitigation recommendation                      | End-to-end workflow orchestration                              | (1) Coordinate information exchange between agents; (2) Share contextual information among agents; (3) Manage the prediction-detection-mitigation workflow.                                                                                                 |

summarizing the inputs, outputs, primary roles, and key tasks of each agent and the coordination layer.

At a high level, the framework operates in three sequential stages. First, the Prediction Agent observes a history of packets belonging to a given network flow and forecasts the next packet, while validating its plausibility against prior context to filter inconsistent predictions. Second, the Detection Agent processes both the forecasted packet and real-time flow data, combining flow-level classification for high-confidence detection with packet-level analysis enriched by flow context for early threat identification. Third, upon detecting malicious activity, the Mitigation Agent maps the attack category to MITRE ATT&CK techniques and retrieves appropriate

countermeasures from the D3FEND knowledge graph. This design yields a unified, lightweight, and proactive defense strategy suitable for dynamic and resource-constrained 6G-enabled IoT environments.

To support agent reasoning, network traffic is represented through structured packet-level and flow-level datasets that preserve temporal dependencies and protocol semantics, serving as a common input space for both prediction and detection.

1) **PACKET-LEVEL REPRESENTATION FOR PREDICTIONS**  
IoT network communications are modeled as ordered packet sequences associated with individual traffic flows. Instead

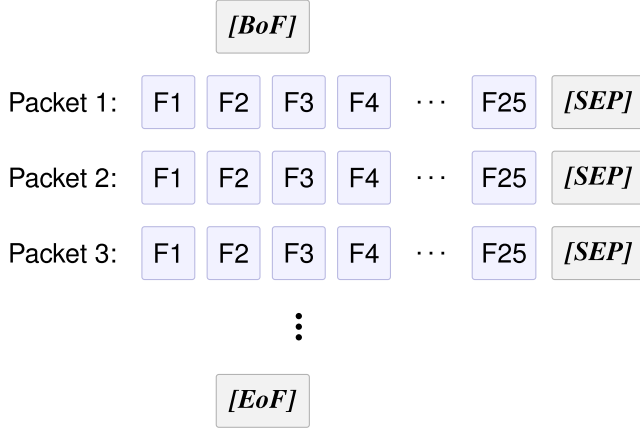


**FIGURE 2.** Textual packet-flow representation used in the  $D_F$  forecasting dataset. Each packet is transformed into a sequence of feature tokens separated by [SEP] markers, with [BoF] (resp. [EoF]) denoting the start (resp. end) of the flow.

of analyzing packets independently or relying solely on aggregated flow statistics, our framework preserves the chronological progression of packets within each connection to capture dependencies. This choice is motivated by the observation that attack patterns often manifest as specific sequences of packet headers, flags, or timing anomalies that are only detectable when temporal context is preserved.

Formally, an “observed traffic flow” (See the Input in Fig. 1) is represented as  $F = (P_1, P_2, \dots, P_n)$ , where each packet  $P_i$  is encoded as a feature vector  $P_i = \phi(p_i) \in \mathbb{R}^k$ , with  $\phi(\cdot)$  denotes a feature extraction function applied to the raw packet  $p_i$ . The extracted attributes capture protocol information (e.g., TCP flags, IP options), timing dynamics (inter-arrival times), and transport-layer semantics derived from layers 2–4 of the TCP/IP stack. We deliberately exclude payload content to maintain lightweight processing and preserve privacy.

This sequential formulation enables autoregressive modeling of packet streams. Concretely, the forecasting task estimates the conditional probability of the next packet given its preceding context:  $P(P_{t+1} | P_1, \dots, P_t)$ . We then transform packet sequences into a tokenized textual representation suitable for fine-tuning SLMs, as illustrated in Fig. 2, and where we explicitly indicate flow boundaries by inserting sentinel markers [BoF] (beginning-of-flow) and [EoF] (end-of-flow). The parsing stage produces two packet-level datasets:  $D_F$  for next-packet forecasting and  $D_V$  for packet-pair validation, the latter being used to verify the consistency between consecutive packets. Importantly, dataset splits are performed at the flow level to ensure that temporally correlated samples remain within the same subset (training, validation, or testing), thereby preventing any data leakage between splits.

## 2) FLOW-LEVEL AND PACKET-LEVEL REPRESENTATIONS FOR DETECTION

The Detection Agent relies on both packet-level and flow-level representations. We construct two datasets accordingly:

- $D_{1C}$ : a packet-classification dataset used for proactive

detection on forecasted packets. Each instance combines packet-level features with rolling flow statistics computed only from past packets, ensuring temporal causality.
- $D_{2C}$ : a flow-classification dataset used for real-time intrusion detection. Each flow is summarized by 65 aggregated features capturing temporal statistics (e.g., inter-arrival times, packet size variations), TCP characteristics (window sizes, round-trip-time (RTT), sequence/acknowledgment behavior), and IP/ICMP metrics (time-to-live (TTL), options, echo ratios). These representations are designed to maintain low computational complexity while retaining sufficient discriminative power for high-accuracy detection.

## B. PREDICTION AGENT

Building on the packet-level representation introduced above, the “Prediction Agent” anticipates future network traffic by modeling the temporal evolution of packets within each flow. It combines complementary tasks: autoregressive next-packet forecasting, packet-pair consistency validation, and proactive intrusion prediction through forecast-driven analysis of malicious activity before flow completion. In this context, packet forecasting refers to generating future packet representations from historical sequences, while proactive intrusion prediction and detection refer to identifying malicious behavior based on these forecasted packets prior to full flow observation. The rationale for adding validation is that raw autoregressive predictions may occasionally violate protocol semantics (e.g., an impossible TCP flag transition). By filtering predictions through a bidirectional validation model, we increase the reliability of the forecasted packets before they are passed to the Detection Agent.

Both tasks leverage the Transformer architecture and its attention mechanism [49]. Given query, key, and value matrices  $Q$ ,  $K$ , and  $V$ , attention is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (1)$$

where  $d_k$  denotes the dimensionality of the key vectors. This allows the SLM model to dynamically focus on the most relevant packets within the sequence when computing contextual representations.

### 1) NEXT-PACKET FORECASTING

Given a packet sequence  $F = (P_1, \dots, P_t)$ , the forecasting task predicts the next packet  $\hat{P}_{t+1}$  as  $\hat{P}_{t+1} = f_\theta(P_1, \dots, P_t)$ , where  $f_\theta$  denotes an autoregressive SLM parameterized by  $\theta$ . We fine-tune the DistilGPT2 SLM [50] using the dataset  $D_F$ . The input sequences are tokenized packet representations, and the model is trained to minimize the causal language modeling loss:

$$\mathcal{L}_{GPT} = - \sum_{k=1}^N \log P(x_k | x_1, \dots, x_{k-1}; \theta). \quad (2)$$

Through this autoregressive learning process, the model captures temporal dependencies and protocol dynamics across packet sequences. Unlike flow-level methods that wait for flow completion, our forecasting agent can anticipate the

next packet after only a few preceding packets, enabling early detection.

## 2) PACKET-PAIR CONSISTENCY VALIDATION

To ensure that a forecasted packet  $\hat{P}_{t+1}$  is plausible given the actual preceding packet  $P_t$ , we train a binary classifier on the dataset  $D_V$ . Each sample consists of a packet pair  $(P_t, P_{t+1})$  labeled as valid (1) if the pair appears consecutively in the same flow, or invalid (0) otherwise. Invalid pairs are generated by randomly shuffling packets from different flows or by introducing protocol violations.

We implement this classifier by fine-tuning DistilBERT [24], a lightweight bidirectional SLM. The model takes the concatenated representation of the packet pair and outputs a validity probability:

$$\mathcal{L}_{PE} = \sum_{(X,y)} \log P(y | X; \theta) \quad (3)$$

where  $X$  denotes the packet-pair representation,  $y \in \{0, 1\}$  is the ground-truth label, and  $\theta$  represents the model parameters.

By leveraging bidirectional contextual representations, the validation model can detect protocol inconsistencies (e.g., a SYN packet following a FIN packet without handshake), anomalous transitions, or semantic violations that may arise during autoregressive prediction. Only forecasted packets that satisfy this validation step (i.e.,  $v_t \geq \tau_v$ , where  $\tau_v$  is a confidence threshold) are forwarded to the Detection Agent. In our implementation, the threshold  $\tau_v$  is set to the default decision boundary of the binary DistilBERT classifier ( $\tau_v = 0.5$ ). Consequently, forecasted packets with  $v_t \geq 0.5$  are classified as valid and forwarded to the Detection Agent, whereas those with  $v_t < 0.5$  are classified as invalid and discarded. To ensure consistency and reproducibility, the same threshold was used throughout all experiments without further adjustment. This filtering mechanism helps ensure that only semantically consistent forecasted packets are considered during proactive detection.

## C. DETECTION AGENT

The “Detection Agent” analyzes both forecasted packets (after validation) and real-time network flows to identify malicious activities. It combines two perspectives: packet-level inspection for early warning, and flow-level monitoring for robust baseline detection.

### 1) FLOW-LEVEL CLASSIFICATION

We train a lightweight Gradient Boosting Machine (LightGBM) [51] classifier on the flow-level dataset  $D_{2C}$ . Each flow  $F = (P_1, \dots, P_n)$  is summarized into a feature vector of 65 aggregated statistics, including temporal moments, TCP flags distribution, window size evolution, and ICMP echo ratios. LightGBM is chosen for its high efficiency, low memory footprint, and ability to handle mixed categorical-numerical features. Flow-level analysis provides an initial assessment of the traffic (as an attack or benign).

### 2) PROACTIVE PACKET-LEVEL CLASSIFICATION WITH FLOW CONTEXT

For proactive detection on forecasted packets  $\hat{P}_{t+1}$ , we enrich the packet-level classifier with flow context. The motivation is that individual packet features alone are often insufficient to discriminate malicious from benign traffic, especially in the early stage of an attack. We therefore construct a hybrid feature vector:

$$x_{\text{hybrid}, t+1} = [x_{\text{packet}, t+1}, \phi(F_t), \hat{y}_{\text{flow}, t}],$$

where: -  $x_{\text{packet}, t+1} \in \mathbb{R}^N$  comprises packet-level features extracted from  $\hat{P}_{t+1}$ , where  $N$  denotes the number of features (e.g., protocol fields, flags, lengths); -  $\phi(F_t)$  captures flow-level statistics from recent traffic up to time  $t$ , such as rolling means, variances, and min/max values of packet sizes and inter-arrival times (strict temporal causality); -  $\hat{y}_{\text{flow}, t} \in \mathbb{R}^C$  denotes the class probability vector output by the flow-level classifier at time  $t$ , providing a soft behavioral context.

We then train a second LightGBM classifier on this enriched feature space using the dataset  $D_{1C}$ . The model maps the hybrid representation to a packet-level prediction:

$$\hat{y}_{\text{packet}, t+1} = \sum_{m=1}^M f_m(x_{\text{hybrid}, t+1}) \quad (4)$$

where  $f_m \in \mathcal{F}$  are regression trees in the ensemble, and  $\hat{y}_{\text{packet}, t+1}$  denotes the predicted packet-level label at time  $t + 1$ . The final prediction is obtained by aggregating the outputs of all trees in the ensemble.

Each tree  $f_m$  minimizes the residual loss:

$$f_m = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^N \mathcal{L}(y_i, \hat{y}_i^{(m-1)} + f(x_{\text{hybrid}}^{(i)})) \quad (5)$$

where: -  $y_i$  denotes the ground-truth label for sample  $i$ , -  $\hat{y}_i^{(m-1)}$  denotes the ensemble prediction at iteration  $m - 1$ , -  $x_{\text{hybrid}}^{(i)}$  denotes the hybrid feature vector for sample  $i$ .

---

### Algorithm 1 Proactive Packet Classification

---

**Require:** Forecasted packet  $\hat{P}_{t+1}$ , current flow  $F_t$

**Ensure:** Proactive class prediction  $\hat{y}_{\text{packet}, t+1}$

- 1:  $x_{\text{packet}, t+1} \leftarrow \phi_p(\hat{P}_{t+1})$
  - 2:  $c_t \leftarrow \phi_c(F_t)$
  - 3: **if**  $\hat{y}_{\text{flow}, t}$  is not available **then**
  - 4:    $\hat{y}_{\text{flow}, t} \leftarrow q_y(\phi_f(F_t))$
  - 5: **end if**
  - 6:  $x_{\text{hybrid}, t+1} \leftarrow [x_{\text{packet}, t+1}, c_t, \hat{y}_{\text{flow}, t}]$
  - 7:  $\hat{y}_{\text{packet}, t+1} \leftarrow h_\omega(x_{\text{hybrid}, t+1})$
  - 8: **return**  $\hat{y}_{\text{packet}, t+1}$
- 

This hybrid approach allows the packet classifier to benefit from the behavioral richness of flow analysis while preserving the low latency of packet-level inspection. As summarized in Algorithm 1, the procedure first extracts packet-level features from the forecasted packet together with contextual information from the current flow (Lines 1–2). If a flow-level prediction is not already available, the flow

classifier is executed to obtain the corresponding behavioral context (Lines 3–5). These packet features, flow statistics, and flow-level predictions are then combined into a hybrid representation (Line 6), which is processed by the LightGBM classifier to generate the proactive attack prediction (Line 7).

#### D. MITIGATION AGENT AND END-TO-END COORDINATION

The “Mitigation Agent” is responsible for translating detection alerts into structured defense recommendations. Rather than relying on hard-coded rules, we adopt a knowledge-driven approach that leverages two widely adopted cybersecurity ontologies: MITRE ATT&CK for adversarial behavior description and MITRE D3FEND for countermeasure mapping.

The agent operates in two steps. First, given a detected alert  $a_t \in \mathcal{A}$  (e.g., “DDoS-UDP Flood”), we map it to one or more ATT&CK techniques  $t_t$  using a predefined lookup table derived from the CICIOT2023 attack taxonomy. For example, a UDP flood is mapped to technique T1498 (Network Denial of Service). Second, we query the D3FEND knowledge graph [52] using  $t_t$  as the entry point, retrieving a set of countermeasures  $\mathcal{R}_t$ . These countermeasures fall into categories such as *Detect*, *Harden*, *Isolate*, *Deceive*, and *Evict*. The agent then outputs a recommended defense action, e.g., “rate limiting” or “connection reset”, intended to support security analysts or downstream enforcement systems. The proposed framework does not directly execute mitigation actions; rather, it generates structured mitigation recommendations that can be reviewed and enforced by external security platforms, network administrators, or automated response systems. Formally, the mitigation process is:

$$a_t \xrightarrow{\mathcal{M}_1} t_t \xrightarrow{\mathcal{M}_2} \mathcal{R}_t,$$

where  $\mathcal{M}_1$  is the attack-to-technique mapping and  $\mathcal{M}_2$  is the D3FEND retrieval function.

The complete framework operates as a coordinated pipeline. Given packet history  $\mathcal{S}_t = (P_1, \dots, P_t)$  and the current flow  $F_t$ , the Prediction Agent generates a forecasted packet  $\hat{P}_{t+1}$  using the SLM model  $f_\theta$  and validates it with the DistilBERT model  $g_\psi$  to obtain a validity score  $v_t$ . If  $v_t$  exceeds threshold  $\tau_v$ , we proceed to proactive detection; otherwise, the forecast is discarded, and no proactive alert is raised. Meanwhile, flow-level detection runs independently, producing  $\hat{y}_{\text{flow},t}$ . For valid forecasts, we construct the hybrid feature vector  $x_{\text{hybrid},t+1}$  and obtain a proactive packet-level prediction  $\hat{y}_{\text{packet},t+1}$  using the LightGBM classifier  $h_\omega$ . The final alert  $a_t$  is determined using a logical OR fusion rule between flow-level and packet-level predictions. Specifically, an alert is raised if either the flow-level classifier predicts a non-benign class or the packet-level classifier identifies malicious activity from the forecasted packet. If an alert is raised (i.e.,  $a_t \neq \text{Benign}$ ), the Mitigation Agent maps it to ATT&CK techniques  $t_t$  and retrieves the corresponding countermeasures  $\mathcal{R}_t$ . Otherwise, an empty mitigation set is returned. Algorithm 2 summarizes the end-to-end

#### Algorithm 2 Prediction–Detection–Mitigation Workflow

**Require:** Packet history  $\mathcal{S}_t = (P_1, \dots, P_t)$ , real-time flow  $F_t$

**Ensure:** Final alert  $a_t$ , mitigation set  $\mathcal{R}_t$

```

1:  $\hat{P}_{t+1} \leftarrow f_\theta(\mathcal{S}_t)$ 
2:  $v_t \leftarrow g_\psi(P_t, \hat{P}_{t+1})$ 
3:  $\hat{y}_{\text{flow},t} \leftarrow q_\gamma(\phi_f(F_t))$ 
4: if  $v_t \geq \tau_v$  then
5:    $x_{\text{hybrid},t+1} \leftarrow [\phi_p(\hat{P}_{t+1}), \phi_c(F_t), \hat{y}_{\text{flow},t}]$ 
6:    $\hat{y}_{\text{packet},t+1} \leftarrow h_\omega(x_{\text{hybrid},t+1})$ 
7: else
8:    $\hat{y}_{\text{packet},t+1} \leftarrow \emptyset$ 
9: end if
10:  $a_t \leftarrow \mathcal{D}(\hat{y}_{\text{packet},t+1}, \hat{y}_{\text{flow},t})$ 
11: if  $a_t \neq \text{Benign}$  then
12:    $t_t \leftarrow \mathcal{M}_1(a_t)$ 
13:    $\mathcal{R}_t \leftarrow \mathcal{M}_2(t_t)$ 
14: else
15:    $\mathcal{R}_t \leftarrow \emptyset$ 
16: end if
17: return  $(a_t, \mathcal{R}_t)$ 
    
```

coordination among the framework agents. The Prediction Agent first generates and validates a forecasted packet, while the flow-level detector analyzes the current traffic flow (Lines 1–3). If the forecast is considered valid, proactive packet-level classification is performed using the hybrid feature representation (Lines 4–9). The packet-level and flow-level decisions are then fused to produce a final alert (Line 10). When malicious activity is detected, the Mitigation Agent maps the alert to ATT&CK techniques and retrieves the corresponding D3FEND countermeasures (Lines 11–15). Finally, the framework outputs the generated alert together with the corresponding mitigation recommendations (Line 17).

Our design ensures that the framework closes the loop from prediction to mitigation recommendation by generating structured defense guidance derived from ATT&CK–D3FEND reasoning. The framework focuses on recommendation generation rather than autonomous execution of defensive actions.

## IV. EXPERIMENTAL SETUP

This section describes the experimental methodology for evaluating the proposed agentic framework, including dataset preparation, flow extraction, and model training.

### A. PACKET AND FLOW DATASET PREPARATION

We select the CICIOT2023 dataset [53] due to its large-scale and recent representation of IoT network traffic, including both benign and diverse malicious behaviors generated from realistic IoT environments. The dataset provides heterogeneous attack scenarios and rich traffic patterns, making it well-suited for evaluating intrusion prediction and classification models in IoT settings.

**TABLE 2.** Selected attack classes from the CICIoT2023 dataset.

| Attack Category | Attack Classes                                                                        |
|-----------------|---------------------------------------------------------------------------------------|
| Benign          | BenignTraffic                                                                         |
| DoS             | DoS-HTTP_Flood, DoS-UDP_Flood,<br>DoS-TCP_Flood                                       |
| DDoS            | DDoS-HTTP_Flood, DDoS-UDP_Flood,<br>DDoS-TCP_Flood, DDoS-SYN_Flood,<br>DDoS-SlowLoris |
| Spoofing        | DNS_Spoofing, ArpSpoofing                                                             |
| Reconnaissance  | Recon-PortScan, VulnerabilityScan                                                     |

In this study, we consider a subset of attack classes covering several key IoT threat categories. Specifically, the selected classes include major IoT attack families targeted by the proposed framework, namely Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), spoofing, and reconnaissance attacks. The latter were chosen since they represent prevalent and operationally significant threats in IoT environments, and exhibit varied behavioral characteristics relevant to IoT threat modeling. Moreover, the inclusion of both volumetric flooding attacks and stealth-oriented activities enables assessment under different attack dynamics and operational conditions. The selected classes and their corresponding categories are summarized in Table 2. The dataset sizes and class-wise distributions for both benign and attack categories are summarized in Table 6, with further details on the constructed subsets provided in the following subsections.

For packet-level processing, we parse packets from the raw CICIoT2023 PCAP traffic using the packet-forensics mode of the Tranalyzer traffic exporter [54]. This process extracts protocol fields from layers 2–4 of the TCP/IP stack. Features with low variance or high correlation are removed to reduce redundancy, and malformed or empty rows are discarded to ensure data consistency. The resulting packet sequences are tokenized and delimited using sentinel markers [BoF] and [EoF] to explicitly indicate flow boundaries for GPT-based next-packet forecasting.

In parallel, we construct flow-level datasets from the CICIoT2023 PCAP files where packets are grouped into flows using the standard 5-tuple identification (source IP, destination IP, source port, destination port, and transport protocol). For each flow, temporal, statistical, and protocol-based features are aggregated across all packets, forming structured flow representations to be used for LightGBM-based flow-level classification. The standard 5-tuple is also adopted for flow-consistent splitting. Specifically, all packets belonging to the same flow are assigned to the same subset (training, validation, or testing) to prevent data leakage across temporally correlated samples. The dataset is split into 70% training and 30% testing at the flow level, with packet-level random partitioning avoided to ensure flow consistency. This flow-level partitioning is consistently applied across all datasets used by the prediction, detection, and validation agents.

From the processed traffic data, three datasets are constructed to support the different agents of the framework:

- **Packet Forecasting Dataset ( $\mathcal{D}_F$ ):** Ordered packet sequences delimited with [BoF] and [EoF] markers, used for fine-tuning the GPT-based next-packet forecasting model.
- **Prediction Validation Dataset ( $\mathcal{D}_V$ ):** Packet pairs labeled as *consecutive* or *non-consecutive*, used to train the DistilBERT-based packet-pair validation model.
- **Packet & Flow Classification Dataset ( $\mathcal{D}_C$ ):** Normalized and encoded packet and flow feature representations supporting multi-class intrusion classification using LightGBM.

In addition, a separate **Forecasted Packet dataset** is built from GPT predictions to evaluate the proactive detection capability of the framework. For proactive detection, the packet classifier uses a hybrid 46-dimensional feature space combining 25 Tranalyzer-extracted packet features, 8 rolling flow aggregates (mean and standard deviation over a sliding window of recent packets, computed from historical packets only), and 13 flow model predictions providing behavioral context. This design ensures temporal causality, as only previously observed packets contribute to the context for predicting future packets. For real-time detection, we extract 65 aggregated flow features from complete flows, capturing temporal statistics (inter-arrival times, packet sizes), TCP characteristics (window sizes, round-trip-time (RTT), sequence/acknowledgment behavior), and IP/ICMP metrics (time-to-live (TTL), options, echo ratios). The complete feature set for both detection modes is summarized in Table 3.

## B. AGENTIC ORCHESTRATION FRAMEWORK

We conducted all experiments using Kaggle notebooks equipped with two Tesla T4 GPUs (16 GB VRAM each) and 29 GB of system memory. The proposed framework employs a role-based multi-agent orchestration layer implemented using the *CrewAI* framework [55] (v1.10.1), a lightweight Python framework for coordinating role-specialized AI agents through structured task-driven *Crew* workflows.

The orchestration layer coordinates forecasting, detection, and mitigation agents. LLAMA 3.2 1B (Ollama runtime) supports reasoning within each agent’s defined scope, assisting with detection outcome interpretation, decision coordination, and mitigation recommendation generation.

Three specialized agents operate within this framework: (i) a *Prediction Agent* for next-packet forecasting and validation, (ii) a *Detection Agent* for intrusion detection and attack classification, and (iii) a *Mitigation Agent* deriving D3FEND-based countermeasures. Each agent operates within a constrained functional scope defined by its assigned task and the underlying ATT&CK–D3FEND knowledge structure.

This design emphasizes decomposition of the cybersecurity workflow into interacting specialized agents coordinated through structured reasoning and information exchange, rather than unrestricted autonomous reasoning.

**TABLE 3.** Original Feature Sets for Packet-Level and Flow-Level Detection.

| Detection Mode                    | Feature Category          | Description                                                                                     |
|-----------------------------------|---------------------------|-------------------------------------------------------------------------------------------------|
| <b>Proactive Packet Detection</b> | Packet Header Fields      | Source/destination ports, packet length, IP flags, TCP flags, UDP length, IP/L4 checksums       |
|                                   | Timing Features           | Packet inter-arrival time (pktIAT), packet trip time (pktTrip), flow duration                   |
|                                   | TCP Sequence Features     | Sequence/acknowledgment numbers, differences, flow lengths                                      |
|                                   | TCP State Features        | TCP window size, window scale, flow status, anomalies, flags                                    |
|                                   | Flow Metadata             | flow statistics, header description                                                             |
| <b>Real-Time Flow Detection</b>   | Flow Statistical Features | Temporal statistics (mean, std, min, max of IAT and packet sizes), packet/byte rates, duration  |
|                                   | TCP Flow Features         | Window size statistics (min, max, avg, init), RTT statistics, sequence/ack faults, MSS, options |
|                                   | IP/ICMP Features          | TTL statistics, IP options, ICMP echo ratio, ICMP fragment count                                |
|                                   | Flow Metadata             | 5-tuple (implied), protocol, timestamps, packet/byte counts                                     |

**TABLE 4.** Prediction Agent Dataset Distribution.

| Dataset                                    | Train         | Test / Inference |
|--------------------------------------------|---------------|------------------|
| Packet Forecasting ( $\mathcal{D}_F$ )     | 132,202 flows | 49,231 flows     |
| Packet-Pair Validation ( $\mathcal{D}_V$ ) | 135,959 pairs | 29,135 pairs     |
| Forecasts set                              | –             | 65,000 packets   |

### C. PREDICTION AGENT: FORECASTING AND VALIDATION

The Prediction Agent is responsible for modeling packet sequence dynamics and validating predicted packet transitions. It consists of two complementary tasks: (i) next-packet forecasting and (ii) packet-pair validation. We implemented both models using the Hugging Face `transformers` library [56]. Table 4 summarizes the dataset sizes used for next-packet forecasting and packet-pair validation. The packet forecasting dataset is approximately balanced across traffic classes, with train flows per class ranging from 5,202 to 15,000, and inference flows ranging from 2,231 to 5,000.

**Next-Packet Forecasting Task:** We fine-tune the DISTILGPT2 model on the Packet Forecasting Dataset ( $\mathcal{D}_F$ ) to perform the forecasting task. The model learns to predict the next packet  $\hat{P}_{t+1}$  given a sequence of packets ( $P_1, \dots, P_t$ ) using causal language modeling. We represented packet sequences in textual form and delimited with the sentinel markers [BoF] and [EoF].

**Packet-Pair Validation Task:** To verify the coherence of predicted packet transitions, we trained a DISTILBERT model on the Prediction Evaluation Dataset ( $\mathcal{D}_V$ ). The dataset contains packet pairs labeled as *consecutive* or *non-consecutive*. The model learns to classify whether a predicted packet logically follows the previous packet within the traffic sequence or not.

The main hyperparameters used for fine-tuning both models are summarized in Table 5. They were selected to balance between training stability and computational efficiency.

### D. DETECTION AGENT: HYBRID PACKET-FLOW CLASSIFICATION

The Detection Agent employs a dual-model design reflecting the complementary strengths of packet and flow analysis.

**TABLE 5.** Hyperparameters for Prediction Agent Models.

| Hyperparameter      | DistilGPT2              | DistilBERT                 |
|---------------------|-------------------------|----------------------------|
| Model               | DistilGPT2              | DistilBERT-base-uncased    |
| Parameters          | 82M                     | 66M                        |
| Batch Size          | 8 (grad. acc. 4)        | 128                        |
| Epochs              | 3                       | 6                          |
| Learning Rate       | $5 \times 10^{-5}$      | $2 \times 10^{-5}$         |
| Warm-up             | 500 steps               | 0.1 ratio                  |
| Weight Decay        | 0.01                    | 0.01                       |
| Mixed Precision     | FP16                    | FP16                       |
| Evaluation Strategy | None                    | Epoch                      |
| Task                | Next-packet forecasting | Packet-pair classification |

**TABLE 6.** Detection Agent Dataset Distribution.

| Category       | Packet Data ( $D_{1C}$ ) |                |               | Flow Data ( $D_{2C}$ ) |                |
|----------------|--------------------------|----------------|---------------|------------------------|----------------|
|                | Train                    | Test           | Forecasted    | Train                  | Test           |
| Benign         | 35,000                   | 15,000         | 5,000         | 189,741                | 81,318         |
| DoS            | 105,000                  | 45,000         | 15,000        | 70,554                 | 30,218         |
| DDoS           | 189,480                  | 81,200         | 25,000        | 1,096,194              | 469,799        |
| Spoofing       | 24,060                   | 10,312         | 10,000        | 92,847                 | 39,806         |
| Reconnaissance | 98,180                   | 42,077         | 10,000        | 241,877                | 103,691        |
| <b>Total</b>   | <b>451,720</b>           | <b>193,589</b> | <b>65,000</b> | <b>1,691,213</b>       | <b>724,832</b> |

We train the flow model on the flow-level dataset  $D_{2C}$ , which aggregates temporal and protocol-based features across complete flows. For proactive detection, we train the packet classifier on  $D_{1C}$  with its feature space extended using flow context, following the hybrid feature construction described in Subsection IV-A.

Both datasets follow a 70% training, 30% testing split, with class distribution summarized in Table 6. We use a separate forecasted packet dataset of 65,000 samples (5,000 per class) to evaluate proactive detection performance. We determine LightGBM hyperparameters empirically to balance model complexity and generalization. The flow model uses 95 leaves, a maximum depth of 8, 300 estimators,

and L1/L2 regularization of 0.1. The packet classifier leverages a lighter configuration with 31 leaves, depth 7, 1500 estimators, lighter regularization (0.08 for L1/L2), subsampling at 0.85 for both samples and features, and early stopping.

We evaluate our models using standard classification metrics, i.e., Accuracy, Precision, Recall, and F1-score, computed from true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$\text{Precision} = TP / (TP + FP) \quad (7)$$

$$\text{Recall} = TP / (TP + FN) \quad (8)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

In addition to these standard classification metrics, we introduce a system-level evaluation metric to quantify the temporal advantage of the proposed proactive detection mechanism. Specifically, we define a detection ratio that measures the fraction of a malicious flow required before an alert can be raised, given by:

$$R_d = \frac{N_c}{L_f}, \quad (10)$$

where  $N_c$  denotes the minimum packet context required before proactive prediction can be triggered, and  $L_f$  represents the total number of packets in the malicious flow. This metric quantifies the fraction of a flow required before an attack can be identified, where lower values indicate earlier detection relative to full-flow analysis.

### E. MITIGATION AGENT: D3FEND-BASED COUNTERMEASURES

The Mitigation Agent recommends defensive actions for detected threats using the MITRE D3FEND knowledge graph [52]. To ensure standardized reasoning, detected attack classes are first assigned to MITRE ATT&CK techniques, which serve as an intermediate representation of adversarial behavior. In this work, we use D3FEND version 1.3.0, which contains over 874 defensive artifacts. The system is designed to query the D3FEND knowledge graph via its API or a local ontology representation to ensure flexibility and reproducibility.

D3FEND groups defensive techniques into strategy categories such as *Detect*, *Harden*, *Isolate*, *Deceive*, and *Evict*. Once an attack is detected, the Mitigation Agent retrieves relevant countermeasures from the D3FEND knowledge graph and outputs corresponding defensive actions.

## V. RESULTS AND DISCUSSION

This section presents our experimental results evaluating the Prediction Agent (Subsection V-A), the proactive and real-time Detection Agent (Subsection V-B), and the Mitigation Agent (Subsection V-C).

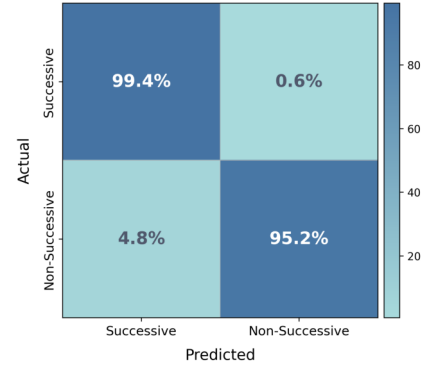


FIGURE 3. Confusion Matrix for Packet-Pair Classification.

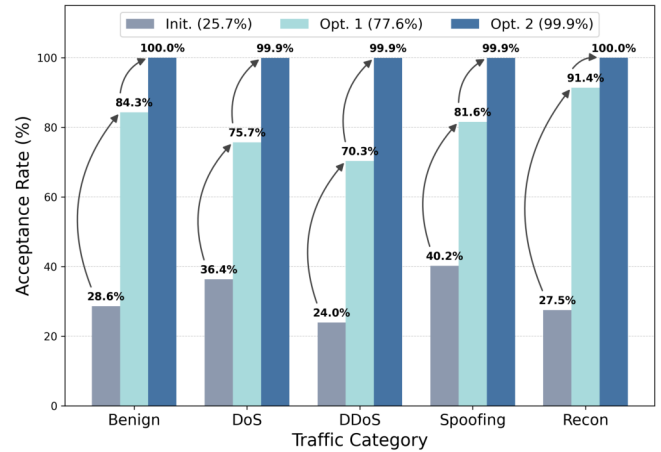


FIGURE 4. Acceptance rate improvement of forecasted packets across attack categories over iterative optimization cycles: initial prediction (Init.), after the 1<sup>st</sup> optimization cycle (Opt. 1), and after the 2<sup>nd</sup> optimization cycle (Opt. 2).

### A. PREDICTION/FORECASTING PERFORMANCE

This subsection evaluates the Prediction Agent's next-packet forecasting and packet-pair validation performances.

First, we present in Fig. 3 the confusion matrix of the DistilBERT-based Packet Pair Classifier (PPC), which classifies packet pairs as successive or non-successive. It achieves an average accuracy of 97.44%, succeeding in well-distinguishing between successive and non-successive packets.

A feedback-driven refinement process is then employed, where forecasted packets generated by DistilGPT2 are iteratively validated and used to guide its optimization, with validation feedback across cycles driving both the selection of additional fine-tuning data and the re-optimization of model hyperparameters. As shown in Fig. 4, the acceptance rate of forecasted packets, defined as the proportion of predicted packet pairs classified as valid successive pairs, increases significantly over three optimization cycles. Starting from 25.74% for the initial prediction, the acceptance rate improves to 77.60% after the first optimization cycle, and reaches 99.90% after the final optimization cycle. Each cycle incorporates feedback from the PPC to correct inconsistencies such as protocol violations and weak temporal

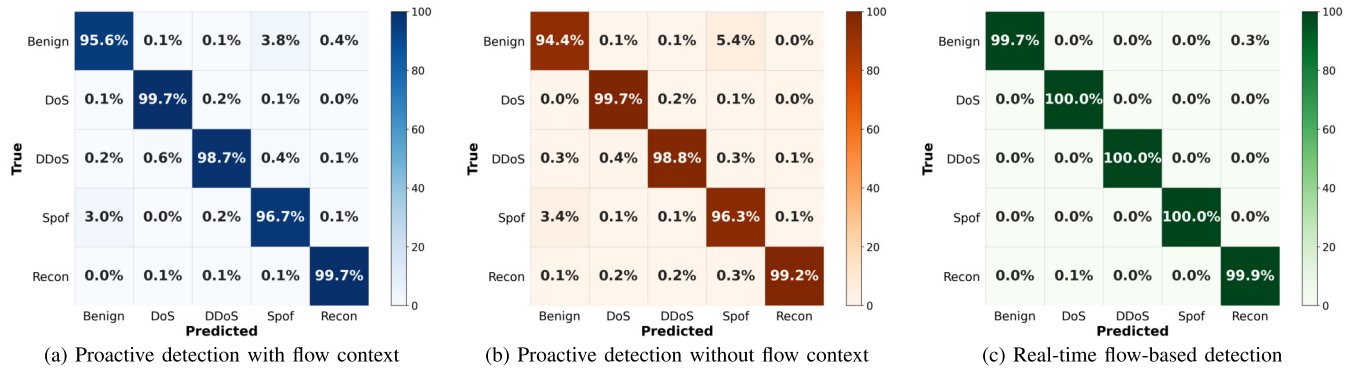


FIGURE 5. Confusion matrices for real-time flow detection and proactive packet detection, with and without flow context.

TABLE 7. Packet-Pair Validation Performance.

| Class          | Prec.  | Rec.   | F1     | Support |
|----------------|--------|--------|--------|---------|
| Successive     | 0.9557 | 0.9949 | 0.9749 | 14,718  |
| Non-Successive | 0.9947 | 0.9539 | 0.9738 | 14,717  |
| Accuracy       | 0.9744 |        |        |         |
| Macro Avg      | 0.9752 | 0.9744 | 0.9744 | 29,435  |
| Weighted Avg   | 0.9752 | 0.9744 | 0.9744 | 29,435  |

dependencies, progressively enhancing prediction quality. This iterative improvement highlights the effectiveness of feedback-guided optimization in producing temporally consistent and valid packet sequences.

### B. PROACTIVE AND REAL-TIME DETECTION PERFORMANCE

In Fig. 5, we present the confusion matrices for three detection settings: (a) proactive packet-level detection with flow context, (b) proactive detection without flow context, and (c) real-time flow-based detection.

Accordingly, we see in Fig. 5a that proactive detection with flow context achieves a strong classification performance, with an average accuracy of 98.24%. The matrix exhibits clear diagonal dominance, indicating that most traffic categories are reliably distinguished at the packet level. Residual errors are primarily concentrated between benign and spoofing traffic, and between UDP-based DoS and DDoS attacks (*DDoS-UDP\_Flood* vs. *DoS-UDP\_Flood*). These classes remain more challenging to distinguish because they exhibit highly similar packet-level characteristics. In particular, spoofing attacks often preserve many legitimate protocol characteristics while manipulating identity-related information, making them difficult to distinguish from benign traffic based solely on individual packet observations. Similarly, UDP-based DoS and DDoS attacks share comparable packet-level patterns, while their distinction becomes clearer when broader flow-level characteristics and behavioral context are considered. Consequently, their discrimination benefits from the temporal and statistical information provided by the flow context. This observation is quantitatively supported by Table 8, where most classes maintain F1-scores

TABLE 8. Proactive Detection Performance: Real vs Forecasted Packets.

| Class             | Real Packets  |               |               | Forecasted   |              |              |
|-------------------|---------------|---------------|---------------|--------------|--------------|--------------|
|                   | Prec.         | Rec.          | F1            | Prec.        | Rec.         | F1           |
| ArpSpoofing       | <b>0.996</b>  | <b>0.998</b>  | <b>0.997</b>  | 0.988        | 0.991        | 0.989        |
| BenignTraffic     | <b>0.999</b>  | <b>0.999</b>  | <b>0.999</b>  | 0.945        | 0.956        | 0.951        |
| DDoS-HTTP_Flood   | 0.997         | <b>0.999</b>  | <b>0.998</b>  | <b>0.999</b> | 0.996        | 0.997        |
| DDoS-SYN_Flood    | <b>1.000</b>  | <b>0.999</b>  | <b>0.999</b>  | 0.999        | <b>0.999</b> | <b>0.999</b> |
| DDoS-SlowLoris    | <b>1.000</b>  | <b>0.999</b>  | <b>1.000</b>  | 0.993        | 0.997        | 0.995        |
| DDoS-TCP_Flood    | <b>1.000</b>  | <b>1.000</b>  | <b>1.000</b>  | 0.999        | 0.999        | 0.999        |
| DDoS-UDP_Flood    | <b>0.964</b>  | <b>0.979</b>  | <b>0.971</b>  | <b>0.964</b> | 0.958        | 0.961        |
| DNS_Spoofing      | <b>0.991</b>  | <b>0.996</b>  | <b>0.994</b>  | 0.981        | 0.993        | 0.987        |
| DoS-HTTP_Flood    | <b>0.997</b>  | 0.994         | 0.996         | 0.996        | <b>0.997</b> | <b>0.997</b> |
| DoS-TCP_Flood     | <b>1.000</b>  | <b>1.000</b>  | <b>1.000</b>  | 0.993        | 0.973        | 0.983        |
| DoS-UDP_Flood     | <b>0.979</b>  | 0.963         | <b>0.971</b>  | 0.961        | <b>0.967</b> | 0.964        |
| Recon-PortScan    | <b>0.999</b>  | <b>1.000</b>  | <b>0.999</b>  | 0.991        | 0.983        | 0.987        |
| VulnerabilityScan | <b>0.999</b>  | <b>0.999</b>  | <b>0.999</b>  | 0.956        | 0.952        | 0.954        |
| Accuracy          | <b>0.9950</b> |               |               | 0.9824       |              |              |
| Macro Avg         | <b>0.9940</b> | <b>0.9940</b> | <b>0.9940</b> | 0.9820       | 0.9820       | 0.9820       |
| Weighted Avg      | <b>0.9950</b> | <b>0.9950</b> | <b>0.9950</b> | 0.9824       | 0.9824       | 0.9824       |

above 0.98, confirming the effectiveness of contextual information in preserving discriminative patterns.

In contrast, Fig. 5b shows that removing flow context leads to a systematic degradation in classification performance, with average accuracy decreasing to 97.90%. The confusion between benign and spoofing traffic becomes more pronounced and bidirectional, indicating that their separation relies heavily on contextual information rather than instantaneous packet features. Reconnaissance-related traffic is also more affected in this setting, suggesting that early-stage packet observations are insufficient to fully capture its behavioral patterns. This degradation is quantitatively confirmed in Table 8, where the F1-score of benign traffic drops significantly from 0.999 (real packets) to 0.951 under forecasted inputs, representing the largest performance decrease among all classes. In comparison, DoS and DDoS categories remain relatively stable, with F1-scores consistently above 0.96, indicating that their high-intensity and structured traffic

TABLE 9. Impact of Flow Context on Proactive Detection (Forecasted Packets).

| Class             | Without Flow Context |              |              | With Flow Context |               |               |
|-------------------|----------------------|--------------|--------------|-------------------|---------------|---------------|
|                   | Prec.                | Rec.         | F1           | Prec.             | Rec.          | F1            |
| ArpSpoofing       | 0.966                | 0.985        | 0.976        | <b>0.988</b>      | <b>0.991</b>  | <b>0.989</b>  |
| BenignTraffic     | 0.935                | 0.944        | 0.939        | <b>0.945</b>      | <b>0.956</b>  | <b>0.951</b>  |
| DDoS-HTTP_Flood   | 0.997                | 0.996        | 0.997        | <b>0.999</b>      | 0.996         | 0.997         |
| DDoS-SYN_Flood    | <b>1.000</b>         | <b>0.999</b> | <b>0.999</b> | 0.999             | <b>0.999</b>  | <b>0.999</b>  |
| DDoS-SlowLoris    | 0.994                | 0.997        | <b>0.996</b> | 0.993             | <b>0.997</b>  | 0.995         |
| DDoS-TCP_Flood    | <b>1.000</b>         | <b>1.000</b> | <b>1.000</b> | 0.999             | 0.999         | 0.999         |
| DDoS-UDP_Flood    | <b>0.969</b>         | 0.955        | <b>0.962</b> | 0.964             | <b>0.958</b>  | 0.961         |
| DNS_Spoofing      | 0.980                | 0.947        | 0.963        | <b>0.981</b>      | <b>0.993</b>  | <b>0.987</b>  |
| DoS-HTTP_Flood    | <b>0.999</b>         | 0.995        | 0.997        | 0.996             | <b>0.997</b>  | <b>0.997</b>  |
| DoS-TCP_Flood     | <b>0.994</b>         | <b>0.985</b> | <b>0.989</b> | 0.993             | 0.973         | 0.983         |
| DoS-UDP_Flood     | <b>0.969</b>         | <b>0.971</b> | <b>0.970</b> | 0.961             | 0.967         | 0.964         |
| Recon-PortScan    | 0.989                | <b>0.983</b> | 0.986        | <b>0.991</b>      | <b>0.983</b>  | <b>0.987</b>  |
| VulnerabilityScan | 0.939                | 0.944        | 0.941        | <b>0.956</b>      | <b>0.952</b>  | <b>0.954</b>  |
| Accuracy          | 0.9790               |              |              | <b>0.9824</b>     |               |               |
| Macro Avg         | 0.9793               | 0.9770       | 0.9781       | <b>0.9819</b>     | <b>0.9816</b> | <b>0.9818</b> |
| Weighted Avg      | 0.9791               | 0.9790       | 0.9790       | <b>0.9825</b>     | <b>0.9824</b> | <b>0.9824</b> |

patterns are inherently more distinguishable at the packet level.

A direct comparison between Fig. 5a and Fig. 5b highlights the critical role of the flow context in reducing the ambiguity of classification. Incorporating flow-level information improves accuracy by 0.34% and significantly mitigates confusion in classes with packet-level characteristics that overlap. As shown in Table 9, this improvement is particularly evident for benign traffic, whose F1-score increases from 0.939 to 0.951 when flow context is incorporated, as well as for DNS spoofing and vulnerability scanning, which exhibit notable gains in recall. These results demonstrate that the flow context effectively compensates for the loss of temporal information in forecasted packets and enhances the robustness of proactive detection.

*Proactive Detection Lead-Time Analysis:* To quantify the temporal advantage of the proposed proactive mechanism, we evaluate the related detection ratio metric  $R_d$  (eq.(10)), where we set  $N_c = 5$  in our implementation, corresponding to the fixed input sequence length used during forecasting model training.

Across all attack categories, the proposed framework consistently detects malicious activity after observing only a small fraction of the flow. In particular, flooding-based attacks (e.g., DDoS-UDP\_Flood and DoS-TCP\_Flood) are typically detected after observing less than 2%–15% of the flow, while more subtle attacks such as reconnaissance and spoofing are detected within approximately 20%–45% of the flow. This corresponds to a substantial reduction (up to 55%–98%) in required observation length compared to conventional flow-based analysis, which requires accumulation of sufficient flow statistics before reliable classification.

Finally, we notice in Fig. 5c that flow-based real-time detection achieves a near-perfect classification performance,

TABLE 10. Flow-Level Real-Time Detection Performance.

| Class             | Prec.         | Rec.   | F1     | Support |
|-------------------|---------------|--------|--------|---------|
| ArpSpoofing       | 0.955         | 0.997  | 0.975  | 2,957   |
| BenignTraffic     | 0.999         | 0.997  | 0.998  | 54,212  |
| DDoS-HTTP_Flood   | 1.000         | 1.000  | 1.000  | 26,048  |
| DDoS-SYN_Flood    | 1.000         | 1.000  | 1.000  | 229,742 |
| DDoS-SlowLoris    | 0.998         | 0.999  | 0.999  | 11,590  |
| DDoS-TCP_Flood    | 1.000         | 1.000  | 1.000  | 71,513  |
| DDoS-UDP_Flood    | 1.000         | 1.000  | 1.000  | 2,104   |
| DNS_Spoofing      | 1.000         | 0.999  | 0.999  | 23,613  |
| DoS-HTTP_Flood    | 1.000         | 0.999  | 0.999  | 12,129  |
| DoS-TCP_Flood     | 1.000         | 0.999  | 1.000  | 5,310   |
| DoS-UDP_Flood     | 0.994         | 0.999  | 0.997  | 2,706   |
| Recon-PortScan    | 0.999         | 1.000  | 0.999  | 69,128  |
| VulnerabilityScan | 1.000         | 0.999  | 0.999  | 97,360  |
| Accuracy          | <b>0.9994</b> |        |        |         |
| Macro Avg.        | 0.9955        | 0.9991 | 0.9973 | 608,412 |
| Weighted Avg.     | 0.9994        | 0.9994 | 0.9994 | 608,412 |

with an overall accuracy of 99.94% and a macro F1-score of 0.9973. The confusion matrix is almost entirely diagonal, confirming that flow-level features capture the intrinsic behavioral signatures of each traffic category. Only negligible confusion is observed, primarily between ArpSpoofing and benign traffic (0.3%) and between DoS-HTTP\_Flood and DoS-UDP\_Flood (0.1%). The small residual confusion between ArpSpoofing and benign traffic can be attributed to the fact that ARP exchanges are also common in normal network operation, resulting in partially overlapping behavioral patterns at the flow level. Likewise, the limited confusion between DoS-HTTP\_Flood and DoS-UDP\_Flood reflects similarities in high-volume attack behavior despite differences in the underlying protocols. This near-perfect behavior is further supported by Table 10, where all classes achieve F1-scores above 0.975, with the majority reaching 0.999 or higher, confirming the strong separability achieved in the flow feature space.

*Computational Efficiency Analysis:* Beyond predictive performance, the proposed detection modules exhibit strong computational efficiency, supporting low-latency operation within Multi-access Edge Computing (MEC)-assisted environments. The flow-based real-time model achieves an average inference latency of 0.046 ms per packet and a throughput of 21,608 packets/s, while maintaining a compact storage footprint of 11.3 MegaBytes (MB). For proactive packet-level detection, the enhanced model incorporating flow context achieves an inference latency of 0.144 ms per packet and a throughput of 6,941 packets/s. Despite the increased feature dimensionality, the model maintains a compact storage footprint of 17.9 MB while preserving sub-millisecond inference performance.

The reported latency and throughput measurements correspond exclusively to the packet-level and flow-level detection modules, including feature extraction, feature processing, and classification with the trained LightGBM models. They do

**TABLE 11.** Attack Mitigation Mapping Based on D3FEND Strategies.

| Strategy                 | Attack Types                         | ATT&CK Tech.         | Countermeasures                               |
|--------------------------|--------------------------------------|----------------------|-----------------------------------------------|
| Harden, Isolate          | ARP Spoofing                         | T1557.002            | ARP inspection, packet filtering              |
| Harden, Isolate, Deceive | DNS Spoofing                         | T1557.001            | DNSSEC, packet filtering, traffic redirection |
| Isolate, Evict           | DDoS (SYN, TCP, UDP), DoS (TCP, UDP) | T1498, T1499         | Connection reset, traffic filtering           |
| Evict, Harden, Isolate   | DDoS-HTTP Flood                      | T1498.002            | Rate limiting, WAF, traffic filtering         |
| Evict, Harden            | SlowLoris, DoS-HTTP Flood            | T1498.002, T1499.004 | Rate limiting, connection reset, WAF          |
| Detect, Harden           | Port Scan, Vulnerability Scan        | T1046, T1595.002     | Network monitoring, access control            |
| –                        | Benign Traffic                       | –                    | No action required                            |

not include the execution time associated with SLM-based packet forecasting, DistilBERT validation, inter-agent coordination, or D3FEND knowledge graph retrieval. Therefore, the reported values should be interpreted as the computational cost of the detection components rather than the complete end-to-end multi-agent framework. Extending the analysis to the full framework remains an important direction for future work.

### C. MITIGATION ANALYSIS

In this work, the evaluation of attack-to-countermeasure mappings aims to validate the correctness of the proposed mitigation reasoning process rather than executing or benchmarking live mitigation actions. Specifically, we assess whether each detected attack category is properly mapped to standardized ATT&CK techniques and subsequently aligned with appropriate D3FEND-based defensive strategies. This evaluation is performed through structured analysis of the generated mappings with respect to established cybersecurity knowledge bases. Hence, the evaluation is knowledge-driven and emphasizes the quality of mitigation recommendations rather than operational performance metrics such as latency or deployment impact. Accordingly, the reported results assess the correctness and relevance of the generated recommendations rather than the effectiveness of their real-world execution.

Table 11 summarizes the mapping between detected attack categories, corresponding ATT&CK techniques, and D3FEND-aligned mitigation strategies. The results demonstrate that the proposed “Mitigation Agent” consistently translates detection outputs into structured defense recommendations using standardized cybersecurity knowledge.

To illustrate the complete mitigation reasoning process, consider a detected *DDoS-HTTP\_Flood* attack. The Mitigation Agent first maps this alert to the ATT&CK technique T1498.002 (Network Denial of Service). Using this technique as input, the D3FEND knowledge graph identifies the corresponding defensive strategies, namely *Evict*, *Harden*, and *Isolate*. These strategies are then translated into concrete recommended actions such as rate limiting, deployment of a Web Application Firewall (WAF), and traffic filtering. This example demonstrates the complete reasoning chain from detected attack class to ATT&CK technique, D3FEND strategy selection, and actionable mitigation recommendation.

The distribution of mitigation strategies across attack categories is illustrated in Fig. 6. It highlights how different

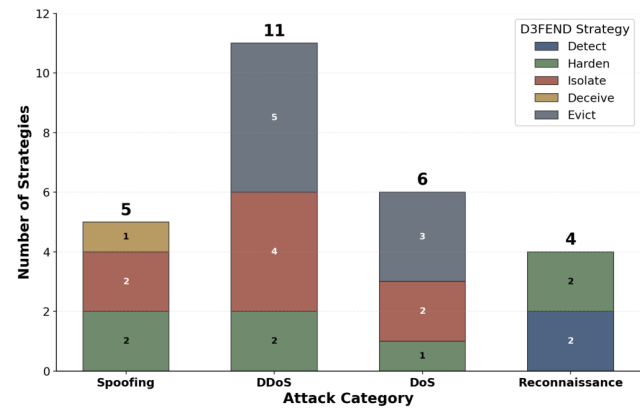


**FIGURE 6.** Distribution of D3FEND-aligned mitigation strategies across different attack categories, illustrating how the Mitigation Agent maps detected attacks to defense actions using ATT&CK-to-D3FEND reasoning.

classes of attacks activate distinct defensive behaviors within the proposed framework. In particular, volumetric attacks (DDoS and DoS) are predominantly mapped to *Evict* and *Isolate* strategies, reflecting their effectiveness in mitigating resource exhaustion through connection reset, traffic filtering, and rate limiting. Reconnaissance activities are associated with *Detect* and *Harden* strategies, emphasizing monitoring and access control. Spoofing attacks exhibit a more diverse mitigation profile, combining *Harden*, *Isolate*, and *Deceive* strategies, reflecting the need for layered defenses such as ARP inspection, packet filtering, and DNS traffic redirection.

### D. COMPARATIVE ANALYSIS AND KEY INSIGHTS

Table 12 compares the proposed framework with recent agentic, traditional, and proactive intrusion detection approaches across multiple criteria, including architecture, model scale, detection granularity, prediction capability, mitigation support, and accuracy.

Recent agentic frameworks such as AgenticCyber [58] and the LLM-based multi-agent system for LAE-IoT [59] (>90%) demonstrate strong detection performance with accuracies 96.8% and over 90%, respectively. However, they remain primarily reactive, relying on flow-level or multi-modal data without incorporating proactive prediction. Their dependence on LLMs also introduces significant computational overhead, limiting their suitability for resource-constrained IoT environments. The single-agent IDS [57] achieves a lower accuracy of 79.5%, lacking the multi-agent coordination and proactive capabilities. Traditional DL

**TABLE 12.** Comparative Analysis of Agentic, Traditional and Proactive IDS Approaches.

| Criterion              | Y. Li [57]   | Roy [58]    | H. Li [59]  | Abbas [31]  | Li et al. [29] | Aydın [46]  | Diaf [17]   | Diaf [16]   | Proposed           |
|------------------------|--------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|--------------------|
| Architecture           | Single Agent | Multi-Agent | Multi-Agent | Traditional | Traditional    | Multi-Agent | Centralized | Centralized | <b>Multi-Agent</b> |
| Language Model Scale   | LLM          | LLM         | LLM         | –           | –              | –           | LLM         | LLM         | <b>SLM</b>         |
| Granularity Detection  | Flow         | Multimodal  | Flow        | Flow        | Flow           | Flow        | Packet      | Packet      | <b>Hybrid</b>      |
| Proactive Prediction   | ✗            | ✗           | ✗           | ✗           | ✗              | ✗           | Partial     | ✓           | ✓                  |
| Fine-Grained Detection | ✗            | ✗           | ✓           | ✓           | ✓              | Limited     | ✗           | ✗           | ✓                  |
| Mitigation Capability  | ✗            | ✓           | ✗           | ✗           | ✗              | ✓           | ✗           | ✗           | ✓                  |
| Accuracy (%)           | 79.5         | 96.8        | >90         | 96.6        | 99.41          | 99.48       | ~98         | ~98         | <b>99.94</b>       |

approaches evaluated on IoT datasets [31] and ensemble models such as LightGBM [29] provide efficient flow-level detection of 96.6% and 99.41%, respectively, yet they remain limited to reactive analysis without forecasting or response mechanisms. Moreover, proactive prediction-based works [16], [17] and multi-agent DDoS defense frameworks [46] demonstrate the feasibility of anticipating or early detecting malicious activity with high accuracy (over 98%), but typically operate without fully integrating detection and mitigation within a unified pipeline or exploiting multi-granularity packet-flow analysis. In contrast, the proposed framework combines a multi-agent architecture with SLM-based forecasting (99.90% validation rate), hybrid packet-flow detection, and knowledge-driven mitigation. It achieves 98.24% accuracy for proactive detection on forecasted packets and 99.94% for real-time flow-based detection on the CICIoT2023 dataset, thus consistently outperforming existing methods.

This comparison highlights the fact that existing methods address individual aspects of intrusion detection, i.e., detection and/or prediction and/or response, whereas the proposed framework integrates these components into a coherent and autonomous pipeline. This integration, combined with multi-granularity analysis and lightweight modeling, results in a scalable, interpretable, and end-to-end efficient cybersecurity defense system.

## VI. CONCLUSION

This paper introduced a multi-agent cybersecurity framework for 6G-enabled IoT networks that integrates SLM-based packet forecasting, multi-granularity LightGBM detection at both packet and flow levels, and D3FEND-aligned mitigation within a coordinated architecture. Experiments on the CICIoT2023 dataset showed that the framework achieves 98.24% accuracy in proactive intrusion detection on forecasted packets and 99.94% accuracy in real-time flow-based detection. By unifying prediction, detection, and mitigation-recommendation generation into a single pipeline, our framework demonstrated the promise of lightweight agentic architectures for proactive and adaptive IoT cybersecurity.

Despite these strengths, the evaluation relied on a single IoT dataset, which, although recent and comprehensive, may not fully represent the heterogeneity of real-world 6G-enabled IoT deployments or the breadth of emerging attack vectors. Moreover, the mitigation module currently

depends on a static D3FEND-based knowledge graph, limiting adaptability to novel or evolving threats, and produces structured recommendations rather than autonomously executed responses. These constraints highlight the need for broader validation and more dynamic mitigation capabilities.

Consequently, future work will focus on enabling automated mitigation with verification mechanisms supported by SLMs, expanding evaluation across multiple IoT datasets to assess generalization, and incorporating continuously updated cybersecurity knowledge to enhance adaptability against emerging threats.

## REFERENCES

- [1] Y. Luo et al., “Securing 5G/6G IoT using transformer and personalized federated learning: An access-side distributed malicious traffic detection framework,” *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1325–1339, 2024.
- [2] DeXpose. (2026). *IoT Hacking Statistics 2026—Global Trends and Key Insights*. Accessed: Mar. 8, 2026. [Online]. Available: <https://www.dexpose.io/iot-hacking-statistics/>
- [3] S. Anthony. (Nov. 2025). *This DDoS Group Just Smashed the Previous Record With a 29.7 Tbps Attack*. Accessed: Mar. 8, 2026. [Online]. Available: <https://www.techradar.com/pro/security/this-ddos-group-just-smashed-the-previous-record-with-a-29-7-tbps-attack>
- [4] I. M. Council. (2023). *IoT Botnet DDoS Traffic Increases Fivefold*. Accessed: Mar. 8, 2026. [Online]. Available: <https://iotm2mcouncil.org/iot-library/news/iot-newsdesk/iot-botnet-ddos-traffic-increases-fivefold/>
- [5] S. Mahmood, M. Gohar, O. A. Khashan, N. Alzahrani, A. Ghani, and F. Al-Turjman, “Securing edge devices in IoT and 6G: A trust-based approach for resource-constrained environments,” *IEEE Open J. Commun. Soc.*, vol. 6, pp. 6568–6610, 2025.
- [6] A. A. Korba, A. Diaf, and Y. Ghamri-Doudane, “AI-driven fast and early detection of IoT botnet threats: A comprehensive network traffic analysis approach,” in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2024, pp. 1779–1784.
- [7] A. A. Korba, A. Diaf, M. A. Bouchiha, and Y. Ghamri-Doudane, “Mitigating IoT botnet attacks: An early-stage explainable network-based anomaly detection approach,” *Comput. Commun.*, vol. 241, Sep. 2025, Art. no. 108270.
- [8] T. Ahmad, D. Truscan, J. Vain, and I. Porres, “Early detection of network attacks using deep learning,” in *Proc. IEEE Int. Conf. Softw. Test., Verification Validation Workshops (ICSTW)*, Apr. 2022, pp. 30–39.
- [9] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, “Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure,” *Sensors*, vol. 23, no. 5, p. 2415, Feb. 2023.
- [10] I. Mutambik, “An efficient flow-based anomaly detection system for enhanced security in IoT networks,” *Sensors*, vol. 24, no. 22, p. 7408, Nov. 2024.
- [11] A. A. Korba, N. E. Karabadi, and Y. Ghamri-Doudane, “Zero-day botnet attack detection in IoV: A modular approach using isolation forests and particle swarm optimization,” 2025, *arXiv:2504.18814*.
- [12] A. A. Korba, S. Sebaa, M. Mabrouki, Y. Ghamri-Doudane, and K. Benatchba, “A life-long learning intrusion detection system for 6G-enabled IoV,” in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2024, pp. 1773–1778.

- [13] N. Gupta, V. Jindal, and P. Bedi, "A survey on intrusion detection and prevention systems," *Social Netw. Comput. Sci.*, vol. 4, no. 5, p. 439, Jun. 2023.
- [14] J. J. Praba, M. Kalaiselvi, S. K. Dasa, N. Malik, R. V. S. Praveen, and N. Udayakumar, "Intrusion detection and security challenges in 6G networks using stochastic graph neural networks," in *Proc. Int. Conf. Inf. Implement., Innov. Technol. (I2ITCON)*, Jul. 2025, pp. 1–6.
- [15] T. E. T. Djaidja, B. Briki, S. M. Senouci, A. Boulouache, and Y. Ghamri-Doudane, "Early network intrusion detection enabled by attention mechanisms and RNNs," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 7783–7793, 2024.
- [16] A. Diaf, A. A. Korba, N. Elislem Karabadi, and Y. Ghamri-Doudane, "Beyond detection: Leveraging large language models for cyber attack prediction in IoT networks," in *Proc. 20th Int. Conf. Distrib. Comput. Smart Syst. Internet Things (DCOSS-IoT)*, Apr. 2024, pp. 117–123.
- [17] A. Diaf, A. A. Korba, N. E. Karabadi, and Y. Ghamri-Doudane, "BARTPredict: Empowering IoT security with LLM-driven cyber threat prediction," in *Proc. GLOBECOM - IEEE Global Commun. Conf.*, Dec. 2024, pp. 1239–1244.
- [18] S. Hamrioui, A. V. Ciocan, R. Djelouah, C. A. M. Hamrioui, and P. Lorenz, "Routing optimization and adaptive defense system for vehicular networks (ROADS-VN)," *Internet Things*, vol. 34, Nov. 2025, Art. no. 101816.
- [19] S. A. Abdulkareem, C. H. Foh, M. Shojafar, F. Carrez, and K. Moessner, "Network intrusion detection: An IoT and non IoT-related survey," *IEEE Access*, vol. 12, pp. 147167–147191, 2024.
- [20] Y. Chen et al., "A survey of large language models for cyber threat detection," *Comput. Secur.*, vol. 145, Oct. 2024, Art. no. 104016.
- [21] S. Jamshidi, N. Shahabi, A. Nikanjam, K. W. Nafi, F. Khomh, and C. Fung, "The role of large language models in IoT security: A systematic review of advances, challenges, and opportunities," *Internet Things*, vol. 34, Nov. 2025, Art. no. 101735.
- [22] M. Yacoubi, O. Moussaoui, and C. Drocourt, "AI for IoMT security: A comprehensive survey of intrusion detection and system architectures," *Internet Things*, vol. 36, Mar. 2026, Art. no. 101869.
- [23] F. Wang et al., "A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with LLMs, and trustworthiness," *ACM Trans. Intell. Syst. Technol.*, vol. 16, no. 6, pp. 1–87, Dec. 2025.
- [24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
- [25] P. Belcak et al., "Small language models are the future of agentic AI," 2025, *arXiv:2506.02153*.
- [26] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous intelligence for complex goals—A comprehensive survey," *IEEE Access*, vol. 13, pp. 18912–18936, 2025.
- [27] S. Sallam, M. El Barachi, and N. Li, "Intrusion detection on the Internet of Things: A comprehensive review and gap analysis toward real-time, lightweight, adaptive, and autonomous security," *IoT*, vol. 7, no. 1, p. 16, Feb. 2026.
- [28] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study," *Appl. Sci.*, vol. 15, no. 4, p. 1903, Feb. 2025.
- [29] W. Chen, H. Yang, L. Yin, and X. Luo, "Large-scale IoT attack detection scheme based on LightGBM and feature selection using an improved Salp swarm algorithm," *Sci. Rep.*, vol. 14, no. 1, p. 19165, 2025.
- [30] S. Alosaimi and S. M. Almutairi, "An intrusion detection system using BoT-IoT," *Appl. Sci.*, vol. 13, no. 9, pp. 1–15, 2023.
- [31] S. Abbas et al., "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Comput. Sci.*, vol. 10, p. e1793, Jan. 2024.
- [32] I. H. Putro, T. Ahmad, and R. M. Ijtihadie, "Enhancing MQTT intrusion detection in IoT using machine learning and feature engineering," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 7855–7884, 2025.
- [33] M. G. Karthik et al., "Energy-efficient intrusion detection with a protocol-aware transformer-spiking hybrid model," *Sci. Rep.*, vol. 16, no. 1, p. 7095, Feb. 2026.
- [34] M. A. Ferrag, L. Maglaras, S. Moschioniannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.
- [35] M. A. Ferrag, M. Debbah, and M. Al-Hawawreh, "Generative AI for cyber threat-hunting in 6G-enabled IoT networks," in *Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud Internet Comput. Workshops (CCGridW)*, May 2023, pp. 16–25.
- [36] M. Assiri, "Artificial intelligence-based intrusion detection and secure communication model for sustainable 6G-IoT networks," *Sci. Rep.*, vol. 16, no. 1, p. 12662, Mar. 2026.
- [37] Y. Li, Z. Xiang, N. D. Bastian, D. Song, and B. Li. (2025). *IDS-Agent: An LLM Agent for Explainable Intrusion Detection in IoT Networks*. [Online]. Available: <https://openreview.net/forum?id=uuCkC4cmIH>
- [38] M. Algarni et al., "An intrusion detection system based on lightweight BERT with meta-classifier for Internet of Things environments," *PeerJ Comput. Sci.*, vol. 12, p. e3590, Feb. 2026.
- [39] F. Adjewa, M. Esseghir, and L. Merghem-Boulahia, "From edge transformer to IoT decisions: Offloaded embeddings for lightweight intrusion detection," *Sensors*, vol. 26, no. 2, p. 356, Jan. 2026.
- [40] R. Prasad, B. Lokesh, D. S. Nivedini, P. P. Singh, A. Maurya, and A. Singh, "Sink vulnerability type prediction using small language model (SLM)," in *Proc. Int. Conf. Cognit. Comput. Eng., Commun., Sci. Biomed. Health Informat. (IC3ECSBHI)*, Jan. 2025, pp. 496–501.
- [41] M. Salayma, "Risk and threat mitigation techniques in Internet of Things (IoT) environments: A survey," *Frontiers Internet Things*, vol. 2, Jan. 2024, Art. no. 1306018.
- [42] B. R. Maddireddy and B. R. Maddireddy, "The role of reinforcement learning in dynamic cyber defense strategies," *Int. J. Adv. Eng. Technol. Innov.*, vol. 1, no. 2, pp. 267–292, 2024.
- [43] A. N. Kalejaiye, "Reinforcement learning-driven cyber defense frameworks: Autonomous decision-making for dynamic risk prediction and adaptive threat response strategies," *Int. J. Eng. Technol. Res. Manag.*, vol. 6, no. 12, pp. 92–111, 2022.
- [44] S. Shen, C. Cai, Y. Shen, X. Wu, W. Ke, and S. Yu, "Joint mean-field game and multiagent asynchronous advantage actor-critic for edge intelligence-based IoT malware propagation defense," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 4, pp. 3824–3838, Jul. 2025.
- [45] Y. Zhou, G. Cheng, K. Du, Z. Chen, T. Qin, and Y. Zhao, "From static to adaptive defense: Federated multi-agent deep reinforcement learning-driven moving target defense against DoS attacks in UAV swarm networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 12, pp. 3407–3421, 2026.
- [46] H. Aydın, G. Z. G. Aydın, A. Sertbaş, and M. A. Aydın, "Internet of Things security: A multi-agent-based defense system design," *Comput. Electr. Eng.*, vol. 111, Nov. 2023, Art. no. 108961.
- [47] A. Tellache, A. Mokhtari, A. A. Korba, and Y. Ghamri-Doudane, "Multi-agent reinforcement learning-based network intrusion detection system," in *Proc. NOMS - IEEE Netw. Operations Manage. Symp.*, May 2024, pp. 1–9.
- [48] A. Tellache, A. A. Korba, A. Mokhtari, and Y. Ghamri-Doudane, "Collaborative privacy-preserving network intrusion detection: A federated multi-agent reinforcement learning approach," *Comput. Commun.*, vol. 256, Jul. 2026, Art. no. 108565.
- [49] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [50] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
- [51] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3149–3157.
- [52] MITRE Corporation. (2021). *MITRE D3FEND Knowledge Graph*. Accessed: Mar. 2026. [Online]. Available: <https://d3fend.mitre.org/>
- [53] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.
- [54] S. Burschka and B. Dupasquier, "Tranalyzer: Versatile high performance network traffic analyser," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.
- [55] (2024). *CrewAI: Framework for Orchestrating Role-Playing, Autonomous AI Agents*. [Online]. Available: <https://github.com/crewAIInc/crewAI>
- [56] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Syst. Demonstrations*, 2020, pp. 38–45.
- [57] Y. Li et al., "IDS-agent: An LLM agent for explainable intrusion detection in IoT networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 1–20.
- [58] S. Roy, "AgenticCyber: A GenAI-powered multi-agent system for multimodal threat detection and adaptive response in cybersecurity," 2025, *arXiv:2512.06396*.
- [59] H. Li et al., "Multi-agent collaborative intrusion detection for low-altitude economy IoT: An LLM-enhanced agentic AI framework," 2026, *arXiv:2601.17817*.