Real-Time Intrusion Detection for IoMT with In-Network Inference on SmartNICs

Aristide Tanyi-Jong Akem[†] and Noa Zilberman[§]
[†]University of Southampton, United Kingdom, [§]University of Oxford, United Kingdom
a.t-j.akem@soton.ac.uk, noa.zilberman@eng.ox.ac.uk

***Abstract*—Internet of Medical Things (IoMT) systems constitute safety-critical networking environments that remain vulnerable to cyber threats. Existing intrusion detection systems typically rely on off-path processing at the edge, fog, or cloud, resulting in increased detection latency and delayed response, which can adversely impact timely intervention in patient-critical scenarios. P4-programmable SmartNICs enable placing machine learning inference directly in the network data path for low-latency, on-path inference without reliance on external processing. In this paper, we present a SmartNIC-based intrusion detection system based on machine learning inference, running entirely on the NIC data plane. Our design builds on a stateless binary decision tree mapped onto the SmartNIC match-action pipeline, enabling per-packet classification entirely in the fast path. We implement our solution in P4 on an Intel IPU and evaluate it using two IoMT datasets. Results show that our approach reduces latency by 10× compared to a host-based system, providing end-to-end latency similar to L2 forwarding, while achieving up to 99% detection accuracy and enabling timely in-network intrusion detection.**

***Index Terms*—In-network inference, SmartNIC, intrusion detection, IoMT, machine learning, P4**

I. Introduction

The Internet of Medical Things (IoMT) is transforming modern healthcare by enabling continuous patient monitoring, remote diagnostics, and real-time clinical decision-making through interconnected medical devices [1]. By integrating wearable, implantable, and hospital-based systems with networked infrastructures, IoMT improves healthcare efficiency and supports early detection of critical conditions. However, this increased connectivity expands the attack surface of healthcare systems, exposing sensitive data and safety-critical operations to cyber threats [2], [3]. Unlike conventional IoT deployments, IoMT systems operate in high-stakes environments where network disruptions or compromised data can directly affect clinical decisions and patient health. Moreover, IoMT applications often involve continuous sensing and feedback loops, where timely responses are required based on collected patient data. These characteristics impose stricter requirements on reliability and latency, making effective intrusion detection essential for secure IoMT operation [4].

Despite substantial progress, existing intrusion detection systems (IDS) for IoMT remain limited by their deployment model. Most approaches rely on machine learning (ML) and deep learning (DL) techniques deployed at the edge/fog, or cloud, where IoMT traffic is aggregated and analyzed [5]–[8]. While these methods achieve high detection accuracy, they introduce fundamental limitations. First, detection is typically *off-path*, requiring traffic to be exported from the forwarding path to external processing entities such as the host or cloud, introducing additional latency, which is undesirable for time-sensitive healthcare applications. Second, using complex ML/DL models imposes significant computational overhead, limiting their deployment in constrained environments. Third, detection is often performed after traffic aggregation, delaying response and limiting the potential for real-time mitigation. These limitations highlight a gap between current IDS designs and the requirements of IoMT environments.

In this paper, we show that moving intrusion detection into the network data plane using SmartNICs enables a new design point for IoMT security. We propose an IDS deployed at the IoMT edge/fog layer, where SmartNIC-enabled servers act as programmable aggregation points between medical devices and cloud services. Leveraging the P4 data-plane programming language [9], we implement *on-path* intrusion detection at a per-packet granularity directly within the packet processing pipeline, eliminating traffic redirection and enabling immediate response. Our design employs a lightweight ML model tailored to data-plane constraints, including limited memory, restricted operations, and feature availability in the fast path, building on recent advances in in-network ML inference [10]–[12]. In addition, executing detection on the SmartNIC provides isolation from the host system and enables efficient offloading of network security functions [13], [14].

Our approach differs from prior IoMT IDS research along two key dimensions: *placement* and *system design*. Existing ML/DL-based IDS solutions primarily focus on improving detection accuracy while assuming off-path deployment at the edge/fog, or cloud. In contrast, we focus on model placement, showing that executing IDS directly in the SmartNIC data plane enables low-latency detection and immediate mitigation. Compared to software-based approaches, which are constrained by CPU processing limits and incur overhead at high packet rates, our SmartNIC-based design provides line-rate processing by keeping the entire detection pipeline on-path. While prior work has explored SmartNIC-based intrusion detection and in-network classification [15]–[18], these efforts either execute inference off-path on on-NIC cores, rely on models or features incompatible with data plane constraints, or target general network environments without IoMT-specific evaluation. This work bridges this gap by designing and evaluating a SmartNIC-based IDS tailored to IoMT deployments.

The main contributions of this paper are as follows:

- We identify a key limitation of prior IoMT IDS solutions, which rely on off-path processing at the edge/fog or

cloud, incurring additional latency and overhead that hinder real-time responsiveness. We address this by proposing a SmartNIC-based IDS architecture enabling fully on-path intrusion detection within IoMT fog servers.
- We design a lightweight ML-based IDS tailored to programmable data plane constraints. We train a stateless binary decision tree using header-derived features and map it to match-action tables for deployment on a SmartNIC, enabling per-packet inference at line rate.
- We implement the proposed solution in P4 on an off-the-shelf Intel IPU and evaluate it using representative IoMT traces, achieving 96–99% detection accuracy while maintaining near L2 forwarding latency and reducing latency by 8–10× compared to host-based approaches.

The rest of this paper is structured as follows. §II reviews background and related work, §III presents the system design, and §IV describes the experimental setup. §V then evaluates system performance and presents results, while §VI discusses several trade-offs. The paper is concluded in §VII.

II. Background and related work

A. IoMT intrusion detection

Previous work has explored conventional and ML-based intrusion detection for IoMT, leveraging both classical and deep learning models for identifying malicious traffic patterns as highlighted in recent surveys [2]–[4]. Most existing approaches adopt edge [19], [20] or fog/cloud-based architectures [5]–[8], where traffic is collected and analyzed outside the packet processing path. These systems often rely on computationally intensive models, such as deep neural networks or ensemble methods, for higher accuracy [21]–[24].

While effective in offline or centralized settings, such designs introduce additional processing overhead and latency. This limits their suitability for real-time detection and rapid response in IoMT systems, where continuous data streams support time-sensitive healthcare services. The limitation motivates the need for more efficient low-latency IDS solutions.

Compared to generic IoT systems, IoMT imposes stricter domain-specific requirements [2]. In healthcare settings, intrusion detection must achieve high reliability, as inaccurate predictions may adversely affect patient monitoring, clinical decision-making, and overall patient safety. Furthermore, IoMT deployments involve heterogeneous communication technologies, with traffic typically aggregated and translated into IP at gateway nodes [1]. Therefore, IDS mechanisms must operate on normalized traffic while remaining lightweight enough for resource-constrained edge/fog deployment.

B. SmartNIC architecture

SmartNICs combine high-speed packet processing with onboard compute resources, enabling network functions to be offloaded from the host while reducing CPU load and improving packet-handling efficiency. However, SmartNICs are not architecturally uniform. As summarized in recent surveys [13], some platforms provide a fully on-path packet-processing pipeline, where traffic is handled directly in the fast path, while others decouple the fast data path from on-NIC general-purpose cores, requiring packets to be steered to the cores when more flexible processing is needed. This distinction is important: fully on-path designs support deterministic packet handling at line rate, whereas core-centric designs offer greater programmability but typically incur additional overhead and are often complemented by software acceleration frameworks such as DPDK or eBPF/XDP [15], [25].

Representative platforms illustrate this diversity, including NVIDIA BlueField Data Processing Units (DPU) [26], Netronome Agilio SmartNICs [27], and Intel Infrastructure Processing Units (IPU) [28], the latter providing a P4-programmable packet-processing pipeline alongside on-NIC general-purpose cores. These systems feature fast-path pipelines and programmable cores, enabling deployment models that balance low-latency on-path detection with more flexible off-path processing, or a combination of both.

C. In-network machine learning

Programmable data planes enable network devices to execute packet processing logic at line rate using abstractions such as match-action (M/A) pipelines [9]. This capability has enabled in-network ML, where inference is performed directly in the data plane, avoiding the need to offload traffic to the control plane or external processing systems and thereby reducing latency [10], [11]. However, data plane execution is constrained by low available memory, limited support for mathematical operations, and bounded pipeline depth.

Due to these constraints, most in-network ML systems rely on tree-based models and their ensemble variants, which can be efficiently mapped to M/A tables, enabling tasks such as traffic classification and anomaly detection at line rate [12], [29]–[33]. More recent efforts have also explored deploying more complex models, such as neural networks, within programmable switch pipelines [34]. While effective, these approaches are typically designed for high-capacity switch architectures and often require multi-stage pipelines, limiting their applicability in more constrained fast-path environments.

More recently, SmartNICs have emerged as an alternative platform for in-network ML, enabling inference to be performed closer to end systems. Prior work has explored ML acceleration on SmartNICs for traffic classification and security applications, including intrusion detection and anomaly detection [15]–[18]. These systems leverage on-NIC compute resources or programmable pipelines to offload inference from the host. However, many such approaches operate off-path on general-purpose cores or rely on complex models that are not directly compatible with strict data plane constraints. To the best of our knowledge, none of these works is designed for, or evaluated in IoMT environments.

D. Gaps and proposed approach

Existing IoMT IDS solutions predominantly rely on off-path processing at the edge, fog, or cloud, introducing additional latency and limiting their ability to support immediate detection and response [5]–[8]. In parallel, in-network ML

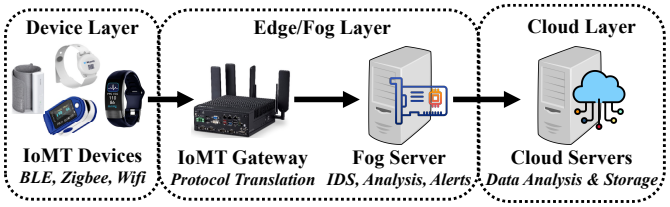


Figure 1: Overview of the IoMT architecture. Wearable and ambient medical devices collect patient data and transmit it via wireless technologies such as Bluetooth, Zigbee, or Wi-Fi. An IoMT gateway translates these protocols into IP traffic for network forwarding. A SmartNIC-enabled fog server performs packet classification for in-network-ML-based intrusion detection and alert generation, after which filtered traffic is forwarded to the cloud for further analysis or storage.

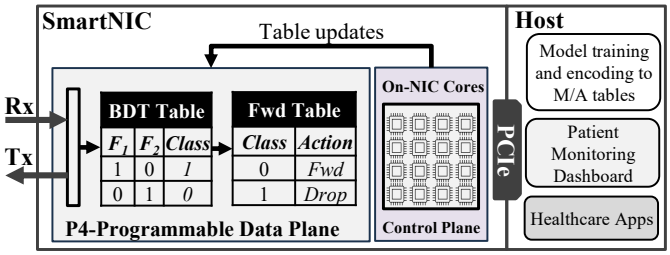


Figure 2: SmartNIC-based IDS architecture. Incoming packets are processed in the P4-programmable data plane, where extracted features are matched against a binary decision tree (BDT) encoded as a match-action table for packet classification. The resulting class determines the forwarding action (*e.g.*, forward or drop). Model training is performed on the host, while on-NIC cores execute control-plane functions.

demonstrates that inference can be executed at line rate within programmable data planes; however, these solutions are largely deployed on switches and target generic networking tasks, making them less suitable for IoMT deployments where traffic is aggregated at gateway-adjacent fog infrastructure [29]–[32].

Recent SmartNIC-based IDS approaches typically execute inference off-path on on-NIC cores or rely on models and features that are not compatible with strict data plane constraints, and are not evaluated in IoMT settings [15]–[18]. As a result, there remains a gap in enabling practical, on-path IDS on SmartNICs that satisfies IoMT requirements for low-latency, resource-efficient, and deployment-aware operation.

This work addresses these gaps by enabling on-path ML inference directly within the SmartNIC data plane for IoMT IDS. We design a lightweight tree-based model that operates on normalized IoMT traffic and fits within SmartNIC resource constraints, allowing real-time packet classification and immediate mitigation without diverting traffic off-path, and demonstrate its effectiveness on representative IoMT datasets.

III. System design

A. System overview

The considered IoMT architecture follows a three-layer model in which medical devices collect patient data and communicate via heterogeneous wireless technologies through a gateway before reaching a fog server, as shown in Figure 1. The gateway performs protocol translation and aggregation, exposing the resulting traffic as standard IP packets. As a result, the SmartNIC operates on network traffic, with healthcare data within packet payloads or application-layer protocols.

Within this architecture, the SmartNIC-enabled fog server is deployed at aggregation points, such as hospital edge servers or regional infrastructure serving multiple healthcare facilities. It is positioned directly in the data path between the gateway and the wider network, ensuring that all device traffic passes through this inspection point after protocol normalization. The intrusion detection model runs on the SmartNIC data plane, enabling line-rate packet classification and action without incurring the additional latency of off-path or cloud processing, which is important for meeting the real-time requirements of healthcare applications that are critical for patient safety.

From a threat perspective, adversaries may originate both externally and internally. Compromised IoMT devices may participate in attacks such as denial-of-service, while attackers can inject malicious traffic targeting devices or services. Such traffic traverses the gateway and is visible at the SmartNIC, enabling early detection and filtering before it propagates further into the infrastructure or reaches cloud services.

B. SmartNIC-based IDS architecture

The SmartNIC-based IDS architecture illustrated in Figure 2 enables on-path packet classification by mapping a lightweight ML model onto a SmartNIC data plane. We assume a SmartNIC with a P4-programmable pipeline, allowing packet parsing, feature extraction, and classification to be executed directly on the fast path. The workflow spans model preparation on the host and inference on the SmartNIC.

Model preparation is performed on the host, where relevant features are selected from historical network traffic and binarized to align with the constraints of data plane execution. Due to limited memory, bounded pipeline stages, and restricted support for arithmetic operations, we adopt a binary decision tree (BDT) model. The trained model is compiled into a match-action (M/A) table, where each root-to-leaf path is encoded as a ternary match rule, enabling the decision process to be implemented within a single table.

A P4 program implementing the classification and forwarding logic is compiled and deployed on the SmartNIC. The on-NIC cores execute a control program that populates the BDT table with the corresponding M/A table entries derived from the trained model. At runtime, incoming packets (Rx) are processed by the P4 pipeline, where the parser extracts header fields as stateless per-packet features. Restricting features to header fields avoids payload inspection, reducing exposure of sensitive data and helping preserve patient privacy. These features are matched against the BDT table to assign a classification result, as shown in Figure 2, with inference performed entirely on the data plane without diverting packets off path.

Based on the classification result, packets are either forwarded, dropped, or redirected for further analysis. Benign traffic is transmitted normally (Tx), while malicious traffic can be filtered or escalated to more advanced detection systems. The host could also concurrently run healthcare applications that consume the forwarded traffic, such as patient monitoring services or alerting mechanisms that notify medical personnel.

C. Model training and mapping

As in prior work on in-network ML inference [10]–[12], model training and encoding are performed outside the data plane, either on the host or an external ML server.

1) Model training: Raw traffic traces are first processed to extract stateless features from IP and transport-layer headers using Tshark [35]. We consider 16 header-derived features, including IP-level attributes such as packet length, protocol, and time-to-live, and TCP/UDP header fields, ensuring compatibility with stateless per-packet processing in the data plane.

We adopt decision tree (DT) models due to their suitability for programmable data planes, where their logical structure can be efficiently mapped to match-action tables, as demonstrated in prior work [29]–[33]. Using Scikit-learn [36], we first train an unconstrained DT on the full feature set and output feature importance scores, which guide the selection of the smallest feature subset that preserves accuracy.

To further facilitate deployment, selected features are binarized before final training. Each feature is represented as a set of binary variables, where a feature of maximum size N bits is decomposed into N one-bit features [32]. This ensures that all decision thresholds are binary, simplifying their implementation in match-action tables. The final binary DT (BDT) is then trained, with tree growth constrained by limiting the number of leaf nodes to a value determined via grid search to balance classification accuracy and table size.

2) Model mapping: The trained BDT is mapped to M/A table entries by encoding each root-to-leaf path as a ternary match rule. Each path represents a conjunction of constraints over the binarized features and is expressed as a value-mask pair, where mask bits indicate whether a feature bit is constrained or treated as a wildcard. The resulting encodings are partitioned according to the original feature boundaries to produce per-feature ternary matches that can be directly applied to packet features extracted in the data plane. Each leaf node thus corresponds to a single table entry containing the match conditions and the associated class label. To ensure correct matching semantics in the presence of overlapping rules, entries are assigned priorities based on their specificity, with more constrained paths taking precedence. This mapping allows the entire decision tree to be implemented within a single match-action table, enabling classification in one stage.

IV. Experimental setup

Hardware setup. We deployed our solution and experimented on an Intel IPU E2100-CCQDA2, a P4-programmable PCIe 4.0 ×16 SmartNIC with dual 100 GbE ports. It is hosted in a server with an Intel Xeon Silver 4510 CPU (2.4 GHz) and 375 GB of DDR4 RAM. To evaluate end-to-end performance, we use a dual-port Mellanox ConnectX-6 100 GbE NIC on the same server to inject traffic directly into the IPU.

Datasets. We employ two publicly available IoMT intrusion detection datasets with different attack types for evaluation.

1) CIC-IoMT [37]: It is a recent benchmarking dataset capturing traffic from 40 IoMT devices (real and simulated) across protocols such as Wi-Fi, MQTT, and Bluetooth. It includes 18 attack scenarios spanning DoS, DDoS, spoofing, reconnaissance, and MQTT-based attacks, with realistic packet-level collection via network taps. We use the provided train-test splits and perform binary classification (benign vs malicious) by grouping all attacks into a single malicious class.

2) ECU-IoHT [38]: The dataset models an Internet of Health Things (IoHT) environment using traffic from medical sensors such as heart rate and blood pressure devices, with network attacks including ARP spoofing, DoS, port scanning, and smurf. It comprises just over 111,000 instances with a small set of available packet features, making it suitable for lightweight IDS evaluation. We perform binary classification using an 80–20 train–test split for benign vs suspicious traffic.

Implementation details. We implement the intrusion detection models as P4 programs in the IPU data plane. For CIC-IoMT, the deployed model employs 6 features and has 448 leaf nodes, while the ECU-IoHT model uses 5 features and grows to 128 leaves. We compare these models with unrestricted and unbinarized baseline decision tree models that use more features and have no limits on the number of leaves or tree depth. The CIC-IoMT baseline uses 16 features, while the ECU-IoHT baseline uses 7 features. These baselines are representative of host and cloud-based approaches that are not subject to data-plane constraints.

Metrics. We evaluate ML classification performance using the F1 score together with true positive and false positive rates, computed as follows: $\mathrm{F1} = \frac{2TP}{2TP+FP+FN}$, $\mathrm{TPR} = \frac{TP}{TP+FN}$, and $\mathrm{FPR} = \frac{FP}{FP+TN}$. For each metric, we report both the macro and weighted averages.

V. Results

We evaluate the performance of the proposed SmartNIC-based intrusion detection approach in terms of classification accuracy, end-to-end latency, resource usage, and scalability.

A. Classification performance

To measure classification performance, we use `tcpreplay` to replay PCAP traces through the SmartNIC. Tables I and II summarize the intrusion detection performances of the proposed solution in comparison to an unrestricted offline baseline. On CIC-IoMT (Table I), our solution achieves performance close to the offline baseline despite operating under data-plane constraints. The weighted F1 score decreases marginally from 0.9788 to 0.9736, while the TPR for malicious traffic remains high at 0.9689. This indicates that the proposed approach preserves strong detection capability even when executed entirely in the data plane. The main difference

Class	TPR		FPR		F1 Score	
	Baseline	This work	Baseline	This work	Baseline	This work
Benign	0.9895	0.9890	0.0244	0.0311	0.9510	0.9394
Malicious	0.9756	0.9689	0.0105	0.0110	0.9863	0.9827
Macro Avg	0.9825	0.9789	0.0175	0.0211	0.9686	0.9611
Weighted Avg	0.9785	0.9731	0.0135	0.0152	0.9788	0.9736

Table I: Classification performance on the CIC-IoMT dataset

Class	TPR		FPR		F1 Score	
	Baseline	This work	Baseline	This work	Baseline	This work
Benign	0.9960	0.9825	0.0033	0.0054	0.9918	0.9813
Malicious	0.9967	0.9946	0.0041	0.0175	0.9978	0.9950
Macro Avg	0.9963	0.9886	0.0037	0.0114	0.9948	0.9881
Weighted Avg	0.9965	0.9921	0.0039	0.0149	0.9965	0.9921

Table II: Classification performance on the ECU-IoHT dataset

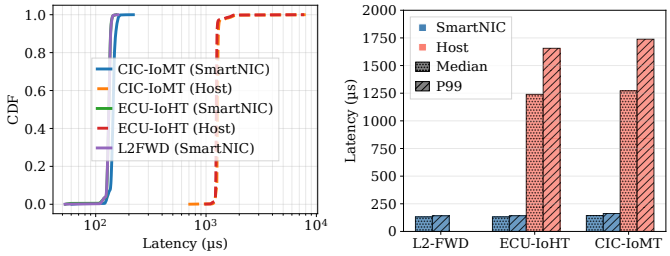
is in the FPR for benign traffic, which increases from 0.0244 to 0.0311, reflecting a moderate rise in false alarms.

On ECU-IoHT, presented in Table II, our SmartNIC-based approach achieves near-parity with the offline baseline. The weighted F1 score remains high at 0.9921 compared to 0.9965 for the baseline, and the TPR for malicious traffic reaches 0.9946. These results demonstrate that for this dataset, the constrained execution environment has minimal impact on detection performance. Similar to CIC-IoMT, a slight increase in FPR is observed, particularly for malicious classification, where it rises from 0.0041 to 0.0175, likely due to the reduced model expressiveness and feature set under data plane constraints, but remaining within acceptable bounds.

The strong performance of simple BDT models suggests that IoMT traffic exhibits separable patterns at the header level, which is advantageous for deployment in constrained environments. Across both datasets, the results highlight a consistent trade-off. Executing inference directly in the SmartNIC fast path introduces a modest increase in false positive rates, while maintaining high detection accuracy and F1 scores. Importantly, this trade-off is achieved without relying on off-path processing in the host or edge infrastructure. As a result, the proposed approach enables real-time detection with minimal performance degradation, supporting its suitability for latency-sensitive IoMT environments.

B. End-to-end latency

We estimate the *end-to-end* latency by measuring the time between packet transmission and classification completion. We compare three scenarios: Layer 2 Forwarding (L2-FWD), our proposed SmartNIC-based approach, and host-based classification on the CPU, which is representative of the state-of-the-art. Packets are generated on the same host using an AF_PACKET socket, with each packet carrying its transmit timestamp embedded in its payload. On the receiving side, packets are captured via a similar socket with kernel-level timestamping enabled. For the SmartNIC-based approach, latency is computed as the difference between the embedded transmit timestamp and the receive timestamp when the classified packet arrives at the host-visible SmartNIC interface.



(a) (b)

Figure 3: (a) Cumulative Distribution Function of end-to-end latency; (b) Median and 99th percentile (P99) of end-to-end latency for SmartNIC-based and host-based classification

Dataset	Setup	Median (μs)	P95 (μs)	P99 (μs)	Max (μs)
L2-FWD	SmartNIC	131.41	138.33	141.61	159.10
CIC-IoMT	SmartNIC	143.88	155.65	162.17	219.30
	Host	1271.05	1305.88	1738.69	7996.76
ECU-IoHT	SmartNIC	131.59	137.98	142.10	153.42
	Host	1239.91	1268.51	1656.15	7734.91

Table III: End-to-end latency statistics for L2 forwarding and two datasets; P95 = 95th percentile and P99 = 99th percentile

For the host-based baseline, packets are similarly received on the host, after which features are extracted, and classification is performed using a pre-trained model with the same features as the SmartNIC model. The timestamp immediately after classification and the embedded transmit timestamp are used to compute the latency. Since packet transmission and reception occur on the same system, timestamps share a common clock, eliminating synchronization errors. The reported latency thus reflects *end-to-end* delay, including forwarding, classification, and, in the host-based case, PCIe transfer and kernel processing, and not isolated per-packet processing time.

Figure 3 and Table III present measurement results across all scenarios. The results show that SmartNIC-based classification achieves tightly bounded latency in the range of approximately 130–145 μs, closely matching the L2 forwarding baseline (L2-FWD) as shown in Figure 3 (a). In particular, the median latency for L2-FWD is 131.41 μs, while SmartNIC-based classification yields 131.59 μs and 143.88 μs for ECU-IoHT and CIC-IoMT, respectively. This indicates that classification is effectively performed at the same rate as packet forwarding, introducing negligible additional latency.

In contrast, host-based classification incurs significantly higher latency, with median values exceeding 1.2 ms across both datasets as shown in Figure 3 (b). Tail latency is also substantially higher, with P99 values above 1.6 ms and maximum latencies approaching 8 ms. Overall, our SmartNIC-based solution improves end-to-end latency by up to 8–10$\times$ while maintaining tightly bounded latency close to the forwarding baseline, with only minor variations across datasets due to differences in feature sets and model complexity. This results in substantially larger improvements in tail latency, reaching up to 30–50$\times$ in worst-case scenarios.

Regarding jitter, SmartNIC-based processing exhibits significantly lower jitter, with tightly bounded latency and minimal

deviation between median and tail values, whereas host-based processing shows substantial variability and long-tail latency due to software and scheduling overheads. These results demonstrate the potential of the proposed solution to enable real-time and low-latency intrusion detection in IoMT systems.

C. Resource overhead

The proposed design is lightweight and fits entirely within the SmartNIC fast path. It uses only two M/A tables: an exact-match table for L2 forwarding and a ternary-match table implementing the BDT. Encoding the BDT within a single table enables inference in a single pipeline step, without requiring additional feature computation, as all features are directly extracted from parsed packet headers. This compact approach contrasts with prior approaches requiring multiple pipeline stages for either tables or feature extraction logic [29]–[32].

The forwarding table contains a single default entry to send traffic to the monitoring port and has a negligible memory footprint. The size of the BDT table scales with the number of tree leaves. For the CIC-IoMT model, the table contains 448 entries of 89 bits each (4.98 KB), while for ECU-IoHT it contains 128 entries of 28 bits each (0.45 KB), both representing a footprint of only a few kilobytes, imposing negligible overhead on on-chip memory resources.

The control plane runs on a single on-board ARM core and is not involved in inference. It is used only for loading and updating table entries, with minimal resource usage of approximately 0.1% CPU and 82 MB memory. The remaining cores remain idle and available for other functions. Overall, these results demonstrate that the proposed solution incurs minimal resource usage on the SmartNIC, allowing it to preserve its primary networking functionalities.

D. Scalability

Our on-path design scales naturally with increasing traffic volume due to its per-packet processing model, which does not require flow aggregation, state synchronization, or memory-intensive tracking structures. This eliminates common scalability bottlenecks associated with stateful network functions.

In addition, the compact representation of the BDT as a single M/A table ensures that scaling to larger tree models primarily impacts table size rather than processing complexity. The IPU supports ternary match keys of up to a few hundred bits, enabling scaling to tens of features. As shown in §V-C, even for the larger CIC-IoMT model, the memory footprint remains within a few kilobytes, indicating that multiple models can be accommodated without significant resource impact.

In contrast, software-based approaches are fundamentally constrained by CPU scalability, where packet processing is subject to scheduling, memory access, and software overheads. Prior work [17] demonstrated that CPU-based solutions can saturate well below line rate, achieving about 20 Gbps at over 90% CPU utilization, highlighting the CPU as a key bottleneck. This limitation is avoided in our design, where inference is executed entirely in the data path. As a result, both packet processing and ML inference are offloaded to the SmartNIC, enabling the data path to sustain line-rate operation while freeing host CPU resources for other tasks.

VI. Discussion

Our experimental results demonstrate the viability of on-path ML inference on SmartNICs for intrusion detection in IoMT networks. We further discuss key aspects related to throughput, deployment, limitations, and future work.

Throughput. The proposed solution operates entirely in the SmartNIC data path, without any control-plane or host involvement during inference. This approach, combined with the stateless nature of the feature set, avoids the need for flow tracking or state maintenance, ensuring deterministic, per-packet, and constant-time processing. It is designed to sustain line-rate operation, similar to switch-based ML inference fully in the data plane [12], [29], [31], [32]. This is further supported by the low resource usage and the latency experiments, where it performs similarly to L2 forwarding, indicating negligible additional processing overhead.

Generality. While our evaluation is conducted on two IoMT datasets, the underlying design is dataset-agnostic. The core pipeline remains unchanged across use cases, with only the feature set and corresponding model parameters varying. As long as classification can be expressed using stateless features derived from packet headers, the approach can be applied directly to a wide range of network monitoring tasks, including intrusion/anomaly detection and traffic classification, without requiring modifications to the system architecture.

Deployment on other targets. Although our implementation targets the Intel IPU, the design principles are not tied to a specific hardware platform. The approach relies on standard P4 constructs, notably ternary M/A tables, supported across a wide range of programmable data plane targets. As such, any SmartNIC or target with a P4-programmable fast path can support the proposed design with minimal modification.

Model choice. The strict memory and processing constraints of the fast path favours compact models such as BDTs that can be encoded efficiently within M/A tables, enabling deterministic, per-packet inference. For more complex use cases, multiple BDTs can be deployed to realize tree-based ensembles such as random forests or gradient-boosted models (*e.g.*, XGBoost), at the cost of increased table size and resource usage, trading off efficiency for potential accuracy gains.

Models that rely on richer features or more complex computations may require collaboration with the on-board ARM cores, where inference or stateful feature computation can be offloaded. Alternatively, model compression techniques such as distillation can be used to approximate complex models with lightweight BDTs suitable for fast path deployment [32].

Model update. If model accuracy degrades over time, *e.g.*, due to the emergence of new threats, updates can be achieved by retraining off-path, *e.g.*, on the ARM cores, and updating the M/A table entries via the control plane. Existing techniques for consistent table updates [39], [40] can be directly applied to our solution for seamless model refresh without disrupting data plane operation or requiring changes to the core design.

Limitations and potential enhancements. The proposed approach is limited to stateless, header-derived features and compact models that can be encoded within M/A tables, and to binary classification, which may restrict applicability to more complex, multi-class, or stateful detection tasks. Supporting richer models may require multi-stage pipelines, inclusion of stateful features, or offloading some tasks to on-board cores. Future work will explore hybrid designs that combine on-path inference with selective offloading, as well as techniques for efficiently mapping more expressive models to the data path while preserving low-latency operation. Additionally, the host-based baseline relies on a kernel networking stack, introducing additional latency. Comparison with optimized software frameworks (*e.g.*, DPDK/XDP) is left for future work.

VII. Conclusion

This paper demonstrated that on-path ML inference on SmartNICs enables accurate, low-latency intrusion detection in IoMT deployments under strict data-plane constraints. A lightweight tree-based model deployed on a programmable SmartNIC pipeline enables real-time attack detection and immediate mitigation without off-path processing. Our results demonstrate a practical, deployable foundation for in-network IoMT security, where timely detection, resource efficiency, and compatibility with heterogeneous traffic are critical, thereby positioning SmartNIC-based data plane processing as an effective alternative to traditional off-path IDS architectures.

Ethics statement: This study uses only publicly available datasets, processes no personal information, and complies with our institutions' guidelines for human-subjects research.

Acknowledgement

This work was partly funded by the Leverhulme Trust and EU Horizon SmartEdge (101092908, Innovate UK 10056403). For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

References

[1] P. He et al. A survey of internet of medical things: technology, application and future directions. *Digit. Commun. Netw. (DCN)*, 2024.
[2] A. Naghib et al. A comprehensive and systematic literature review on intrusion detection systems in the internet of medical things: current status, challenges, and opportunities. *Artif. Intell. Rev.*, 58(4):114, 2025.
[3] A. I. Newaz et al. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *ACM Trans. Comput. Healthcare*, 2(3), July 2021.
[4] A. Si-Ahmed et al. Survey of machine learning based intrusion detection methods for internet of medical things. *Applied Soft Computing*, 2023.
[5] P. Kumar et al. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Computer Communications*, 166:110–124, 2021.
[6] F. Khan et al. A secure ensemble learning-based fog-cloud approach for cyberattack detection in IoMT. *IEEE Transactions on Industrial Informatics*, 19(10):10125–10132, 2023.
[7] A. Berguiga et al. HIDS-IoMT: A deep learning-based intelligent intrusion detection system for the internet of medical things. *IEEE Access*, 13:32863–32882, 2025.
[8] Y. Rbah et al. Deep learning-based fog-cloud approach intrusion detection system in IoMT. *Iraqi Journal for Computer Science and Mathematics*, 6(4):Article 11, 2025.
[9] P. Bosshart et al. P4: programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, July 2014.
[10] R. Parizotto et al. Offloading machine learning to programmable data planes: A systematic survey. *ACM Comput. Surv.*, 56(1), August 2023.
[11] C. Zheng et al. In-network machine learning using programmable network devices: A survey. *IEEE Commun. Surv. Tutor.*, 26(2), 2024.
[12] C. Zheng et al. Planter: Rapid prototyping of in-network machine learning inference. *SIGCOMM Comput. Commun. Rev.*, 54(1), 2024.
[13] E. F. Kfoury et al. A comprehensive survey on SmartNICs: Architectures, development models, applications, and research directions. *IEEE Access*, 12:107297–107336, 2024.
[14] S. Elizalde et al. A survey on security applications with SmartNICs: Taxonomy, implementations, challenges, and future trends. *Journal of Network and Computer Applications*, 242:104257, 2025.
[15] S. Elizalde et al. Accelerated anomaly detection on IoT traffic using SmartNICs. In *Proc. of IEEE GlobeCom*, 2025.
[16] K. Tasdemir et al. An investigation of machine learning algorithms for high-bandwidth SQL injection detection utilising BlueField-3 DPU technology. In *Proc. of IEEE SOCC*, pp. 1–6, 2023.
[17] R. A. Bakar et al. Next-generation intrusion prevention system using hardware-accelerated data processing units (DPUs). In *Proc. of IEEE ICC (ICC Workshops)*, pp. 1438–1443, 2025.
[18] R. Kapoor et al. ML-NIC: accelerating machine learning inference using smart network interface cards. *Front. Comput. Sci.*, 6:1493399, 1 2024.
[19] G. Nagarajan et al. A trust-centric approach to intrusion detection in edge networks for medical internet of thing ecosystems. *Comput. Electr. Eng.*, 115(C), 2024.
[20] M. I. Khalid et al. Lightweight and interpretable edge intelligence ai with intrusion detection for trustworthy cardiac arrhythmia in medical iot. *Scientific Reports*, 16(1):43578, mar 2026.
[21] M. Fouda et al. A novel intrusion detection system for internet of healthcare things based on deep subclasses dispersion information. *IEEE Internet Things J.*, 10(10):8395–8407, 2023.
[22] V. Ravi et al. Deep learning-based network intrusion detection system for internet of medical things. *IEEE Internet Thing Mag*, 6(2), 2023.
[23] S. Khan and A. Akhunzada. A hybrid DL-driven intelligent SDN-enabled malware detection framework for internet of medical things (IoMT). *Computer Communications*, 170:209–216, 2021.
[24] U. Zukaib et al. Meta-IDS: Meta-learning-based smart intrusion detection system for internet of medical things (IoMT) network. *IEEE Internet Things J.*, 11(13):23080–23095, 2024.
[25] S. Rajagopalan et al. Evaluation of programmable packet processing framework using P4 and XDP enabled switches. In *IEEE NetSoft*, 2025.
[26] Bluefield data processing unit (DPU). https://www.nvidia.com/en-gb/networking/products/data-processing-unit/.
[27] Agilio CX SmartNICs. https://netronome.com/agilio-smartnics/.
[28] Intel Infrastructure Processing Unit (Intel IPU). https://www.intel.com/content/www/us/en/products/details/network-io/ipu/adapter-e2100.html.
[29] A. T.-J. Akem et al. Practical and general-purpose flow-level inference with random forests in programmable switches. *IEEE Transactions on Networking*, 33(5):2489–2506, 2025.
[30] G. Zhou et al. An efficient design of intelligent network data plane. In *32nd USENIX symposium on security*, 2023.
[31] B. M. Xavier et al. Map4: A pragmatic framework for in-network machine learning traffic classification. *IEEE Transactions on Network and Service Management*, 19(4):4176–4188, 2022.
[32] G. Xie et al. Empowering in-network classification in programmable switches by binary decision tree and knowledge distillation. *IEEE/ACM Transactions on Networking*, 32(1):382–395, 2024.
[33] A. Angi et al. Routing with ART: Adaptive routing for P4 switches with in-network decision trees. In *Proc. of IEEE GlobeCom*, 2024.
[34] Z. Zhao et al. RIDS: Towards advanced IDS via RNN model and programmable switches co-designed approaches. In *Proc. of IEEE INFOCOM*, pp. 591–600, 2024.
[35] Tshark. https://www.wireshark.org/docs/man-pages/tshark.html.
[36] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.
[37] S. Dadkhah et al. CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT. *Internet of Things*, 28, 2024.
[38] M. Ahmed et al. ECU-IoHT: A dataset for analyzing cyberattacks in internet of health things. *Ad Hoc Networks*, 122:102621, 2021.
[39] M. Zang et al. Toward continuous threat defense: in-network traffic analysis for IoT gateways. *IEEE Internet Things J.*, 11(6), 2024.
[40] H. Yan et al. Linc: Enabling low-resource in-network classification and incremental model update. In *Proc. IEEE ICNP*, pp. 1–12, 2024.