# P4Pir: In-Network Analysis for Smart IoT Gateways

Mingyuan Zang[§], Changgang Zheng[†], Radostin Stoyanov[†],
Lars Dittmann[§], and Noa Zilberman[†]
[§]Technical University of Denmark, [†]University of Oxford
[§]Kgs. Lyngby, Denmark, [†]Oxford, United Kingdom
[§]minza@dtu.dk, ladit@fotonik.dtu.dk, [†]{name.surname}@eng.ox.ac.uk

## ABSTRACT

IoT gateways are vital to the scalability and security of IoT networks. As more devices connect to the network, traditional hard-coded gateways fail to flexibly process diverse IoT traffic from highly dynamic devices. This calls for a more advanced analysis solution. In this work, we present P4Pir, an in-network traffic analysis solution for IoT gateways. It utilizes programmable data planes for in-band traffic learning with self-driven machine learning model updates. Preliminary results show that P4Pir can accurately detect emerging attacks based on retraining and updating the machine learning model.

## KEYWORDS

In-network computing; Machine learning; Security; Internet of Things; P4

## 1 INTRODUCTION

Recent years have seen a widespread deployment of Internet of Things (IoT) devices. IoT gateways are one of the network components that connect IoT devices with the core network, providing functions such as data routing and filtering. Due to dynamic IoT deployments and increasing security threats, IoT gateways are expected to provide traffic analysis and first-line of defense over multi-source traffic inputs [18].

Machine Learning (ML) algorithms are used to enhance the analytical capabilities of IoT gateways [7, 15]. By applying ML models as part of traffic processing, a gateway can learn from traffic patterns and improve its analytical capabilities [9]. However, efficient ML deployment at IoT gateways remains a barrier. In particular, the following challenges exist:

**a) Multi-source input collection:** Current solutions collecting and parsing diverse IoT traffic as the input to ML models are using predefined protocol stacks, either hardware based [13] or software toolkits [17]. When a gateway connects to new devices, running diverse protocols or access technologies, these collection mechanisms are limited and unfit for scalable traffic analytics.

**b) ML inference at the gateway:** ML inference solutions rely on server-based ML frameworks (e.g., Pytorch [14] and Tensorflow [1]) for easy implementation and deployment at IoT gateways. Such frameworks are based on CPU/GPU, where traffic needs to be sent from the network-pipeline to the CPU/GPU for processing, bringing extra overheads. For cases that require large models and high inference accuracy, packet information may be further forwarded to remote servers/clouds for powerful computing, resulting in even longer Round-Trip Times (RTT).

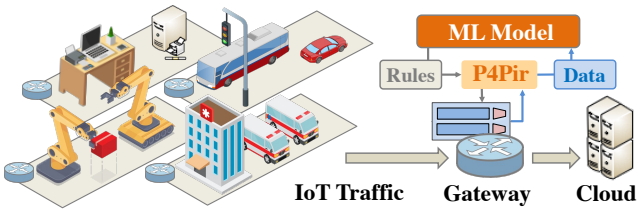


**Figure 1: P4Pir in IoT scenario.**

**c) ML model updates:** To keep the ML model updated with dynamic IoT traffic, two typical methods have been studied: unsupervised online learning [12] and supervised model retraining [4, 10]. The former saves the extra labeling work, but yields lower accuracy and reliability. The latter solutions give reliable learning performance, but their proposed update schemes do not support updates at runtime.

In recent years, the development of programmable data planes (PDP) and in-network computing using the P4 language [2] have provided new opportunities to solve these challenges. Protocol-independent and programmable packet processing pipelines drive in-network analytics, including ML inference [3, 19] and rule updates at runtime [5].

In this work, we present P4Pir, an in-network traffic analysis solution integrated within IoT gateways as shown in Figure 1. With runtime data parsing and data plane inference rules updates at the IoT gateway, P4Pir achieves real-time multi-protocol data collection, in-network ML-based attack mitigation, and runtime model updates. Our preliminary evaluation on the low-cost P4Pi platform [11] shows that P4Pir can improve the accuracy of detecting new attacks by over 50% compared to static ML-based solutions.

## 2 SYSTEM DESIGN

Compared with existing in-network inference solutions, running entirely in the data plane (e.g., [3, 19]), P4Pir engages the control plane for runtime model updates without interrupting the existing processing on the gateway target.

Figure 2 depicts P4Pir workflow. Step ① shows a typical workflow of in-network ML-based detection [20], where a trained model is mapped to Match/Action table rules and P4 code to analyze the arriving traffic.

P4Pir supports common IoT protocol headers defined in P4 (e.g., messaging protocols [17]) and achieves continuous learning beyond existing frameworks by actively collecting traffic features and updating the model, shown as steps ② - ⑦, to detect and mitigate emerging threats. Detailed workflow is illustrated as follows.

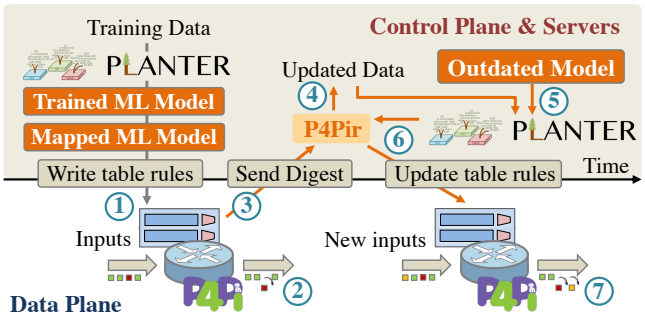


**Figure 2: Block diagram of P4Pir workflow.**

On top of the existing in-network inference solution, as shown in step ②, P4Pir identifies suspicious traffic from incoming traffic. While benign traffic is being forwarded, suspicious traffic will be dropped immediately and logged to the control plane by encapsulating the extracted features in a digest (as in step ③). Based on these digests (labeled by calibration set as in [8]), P4Pir retrains the current model to learn from the new traffic pattern (as in steps ④ & ⑤). A set of new rules will be generated to map the parameters of the new model. Updated rules are then inserted to the data plane and outdated rules are removed (as in step ⑥). With this updated setup, P4Pir is able to learn from newly arriving traffic and mitigate abnormal traffic continuously (as shown in step ⑦).

Two update options are available in P4Pir: parameter updates and feature updates. Considering the changing distribution of arriving traffic, the model can be directly updated by inserting the rules with new parameters and thresholds. When the accuracy shows severe decrement, it probably means the current features can no longer portray the current traffic pattern and new features are needed. Thus, the system can be re-initialized with a new feature extraction and model mapping process.

## 3 PROTOTYPE AND EVALUATION

P4Pir prototype was developed on P4Pi [11], using Raspberry Pi 4 (RPi) Model B with 8GB of RAM, and running P4Pi release v0.0.3 using bmv2 with v1model architecture [16]. For performance evaluation, P4Pir was connected to another RPi as the client and a laptop with an Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz and 16 GB RAM as the server. We used a public IoT dataset *EDGE-IIOTSET* [6] for model training and evaluation. The dataset includes both benign traffic collected from different IoT sensors via diverse IoT protocols and malicious traffic with IoT protocol-related attacks. The attacks are launched in different time slots in a week-worth of data. To evaluate the efficiency of continuous learning from P4Pir, we assume as the initial state that the gateway learns only the DDoS TCP SYN attack on the first day and test the model detection accuracy by replaying new attacks launched in the following days (common IoT attacks: vulnerability scanning/HTTP flooding/UDP flooding). We then compare this baseline's accuracy with the accuracy when P4Pir is deployed to update the model and rules.

**Preliminary results.** Table 1 lists accuracy and F1 score results of P4Pir implemented on Planter-based [20] Decision Tree (DT) and Random Forest (RF). Five L4-based traffic features are used, the DT model is trained with depth of 5 and 1000 leaves, and

| | | SYN | SYN → SCAN | | SYN → HTTP | | SYN → UDP | |
|---|---|---|---|---|---|---|---|---|
| | | Init | Base | P4Pir | Base | P4Pir | Base | P4Pir |
| DT | ACC | 0.995 | 0.460 | 0.998 | 0.360 | 0.999 | NaN | 0.886 |
| | F1 | 0.998 | 0.630 | 0.998 | 0.530 | 0.999 | NaN | 0.939 |
| RF | ACC | 0.999 | 0.994 | 0.997 | 0.340 | 0.998 | NaN | 0.999 |
| | F1 | 0.999 | 0.997 | 0.998 | 0.510 | 0.999 | NaN | 0.999 |

**Table 1: Preliminary results. (Init - Initial state, Base - Baseline, SYN - DDoS TCP SYN attack, SCAN - vulnerability scanning attack, HTTP - HTTP flooding attack, UDP - UDP flooding attack).**

RF model is trained with 5 trees, depth of 5 and 1000 leaves. The results show: 1) DT/RF mapped to the data plane for in-network inference can achieve the same level of accuracy as the baseline performance on a server in [6], reaching more than 90% accuracy and F1 score. 2) P4Pir has benefits in improving DT's performance on vulnerability scanning attacks, given that DT is less scalable than RF. 3) Different attack patterns may result in different levels of accuracy decrement for static models. It might be due to the changing attack attributes and distributions that static models have not learned. When P4Pir is deployed, its update mechanism effectively learns and updates DT/RF parameters to detect new attacks with increased accuracy of 50% or more. Specifically, P4Pir enables a gateway to identify UDP flooding after updates, compared to the baseline when a gateway is not trained with UDP attack. As for the time used for updates, it takes ~0.05s to retrain a model and ~0.45s to update new rules in the data plane.

## 4 CONCLUSION

P4Pir is an in-network ML based analysis solution for IoT gateways, prototyped on the low-cost P4Pi platform. P4Pir enables self-driven learning and ML updates using the interaction between data plane and control plane. Preliminary results in the detection of new attacks demonstrate that P4Pir can scale and detect emerging attacks by retraining and updating in-network models. *Future work* will focus on a distributed deployment of P4Pir and coordinating updates among P4Pir gateways in a federated manner.

## ACKNOWLEDGMENTS

This work was partly funded by the Otto Mønsted Foundation and VMWare.

## REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: Programming Protocol-Independent Packet Processors. *SIGCOMM Comput. Commun. Rev.* 44, 3 (jul 2014), 87–95. https://doi.org/10.1145/2656877.2656890

[3] Coralie Busse-Grawitz, Roland Meier, Alexander Dietmüller, Tobias Bühler, and Laurent Vanbever. 2019. pForest: In-network inference with random forests. *arXiv* (2019). arXiv:1909.05680

[4] Mojtaba Eskandari, Zaffar Haider Janjua, Massimo Vecchio, and Fabio Antonelli. 2020. Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet of Things Journal* 7, 8 (2020), 6882–6897. https://doi.org/10.1109/JIOT.2020.2970501

[5] Yong Feng, Zhikang Chen, Haoyu Song, Wenquan Xu, Jiahao Li, Zijian Zhang, Tong Yun, Ying Wan, and Bin Liu. 2022. Enabling In-situ Programmability in Network Data Plane: From Architecture to Language. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 635–649.

[6] Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras, and Helge Janicke. 2022. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications: Centralized and Federated Learning. https://doi.org/10.21227/mbc1-1h68

[7] Ibbad Hafeez, Markku Antikainen, Aaron Yi Ding, and Sasu Tarkoma. 2020. IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge. *IEEE Transactions on Network and Service Management* 17, 1 (2020), 45–59.

[8] Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. 2017. Transcend: Detecting Concept Drift in Malware Classification Models. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 625–642. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/jordaney

[9] Tran Viet Khoa, Yuris Mulya Saputra, Dinh Thai Hoang, Nguyen Linh Trung, Diep Nguyen, Nguyen Viet Ha, and Eryk Dutkiewicz. 2020. Collaborative learning model for cyberattack detection systems in iot industry 4.0. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[10] Roman Kolcun, Diana Andreea Popescu, Vadim Safronov, Poonam Yadav, Anna Maria Mandalari, Yiming Xie, Richard Mortier, and Hamed Haddadi. 2020. The Case for Retraining of ML Models for IoT Device Identification at the Edge. *CoRR* abs/2011.08605 (2020). arXiv:2011.08605 https://arxiv.org/abs/2011.08605

[11] Sándor Laki, Radostin Stoyanov, Dávid Kis, Robert Soulé, Péter Vörös, and Noa Zilberman. 2021. P4Pi: P4 on Raspberry Pi for Networking Education. *SIGCOMM Comput. Commun. Rev.* 51, 3 (jul 2021), 17–21. https://doi.org/10.1145/3477482.3477486

[12] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. February (2018), 18–21. https://doi.org/10.14722/ndss.2018.23204 arXiv:1802.09089

[13] Arman Pashamokhtari, Norihiro Okui, Yutaka Miyake, Masataka Nakahara, and Hassan Habibi Gharakheili. 2021. Inferring Connected IoT Devices from IPFIX Records in Residential ISP Networks. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. 57–64. https://doi.org/10.1109/LCN52139.2021.9524954

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[15] Arunan Sivanathan, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2020. Managing IoT Cyber-Security Using Programmable Telemetry and Machine Learning. *IEEE Transactions on Network and Service Management* 17, 1 (2020), 60–74. https://doi.org/10.1109/TNSM.2020.2971213

[16] Radostin Stoyanov, Adam Wolnikowski, Robert Soulé, Sándor Laki, and Noa Zilberman. 2021. Building an Internet Router with P4Pi *(EuroP4 '21)*. ACM, New York, NY, USA, 151–156. https://doi.org/10.1145/3493425.3502762

[17] Qinying Wang, Shouling Ji, Yuan Tian, Xuhong Zhang, Binbin Zhao, Yuhong Kan, Zhaowei Lin, Changting Lin, Shuiguang Deng, Alex X. Liu, and Raheem Beyah. 2021. MPInspector: A Systematic and Automatic Approach for Evaluating the Security of IoT Messaging Protocols. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 4205–4222.

[18] Qiao Yan, Wenyao Huang, Xupeng Luo, Qingxiang Gong, and F. Richard Yu. 2018. A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things. *IEEE Communications Magazine* 56, 2 (2018), 30–36. https://doi.org/10.1109/MCOM.2018.1700621

[19] Changgang Zheng, Zhaoqi Xiong, Thanh T Bui, Siim Kaupmees, Riyad Bensoussane, Antoine Bernabeu, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. 2022. IIsy: Practical In-Network Classification. https://doi.org/10.48550/ARXIV.2205.08243

[20] Changgang Zheng, Mingyuan Zang, Xinpeng Hong, Riyad Bensoussane, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. 2022. Automating In-Network Machine Learning. https://doi.org/10.48550/ARXIV.2205.08824