****

**Flow-Based Intrusion Detection on**

**Resource-Constrained Devices**

**Ishan Singh**

**22BCE2608**

**Introduction**

The proliferation of cloud services and IoT devices has **widened the corporate attack surface**, enabling zero-day, *living-off-the-land* and slow-burn attacks that easily bypass signature-based Network IDS (NIDS). Flow-level telemetry—exported as NetFlow/IPFIX records or generated on-device with tools such as Cisco **Joy**—offers a privacy-preserving alternative to payload inspection while retaining rich statistical cues.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)[[4]](#fn4)

The **LUFlow** data set (16 engineered features, weekly refresh) continuously captures production and honeypot traffic inside Lancaster University’s network, auto-labelling each flow as *benign*, *malicious* or *outlier* through threat-intelligence correlation. Its realistic mix of legitimate and emerging threats makes LUFlow an ideal benchmark for **data-driven, real-time IDS** research and motivates the problem addressed here.[[5]](#fn5)[[6]](#fn6)

**Problem Statement:**

*How can we design, benchmark and deploy an intrusion-detection model that meets strict accuracy and latency targets on commodity edge hardware while remaining adaptable to LUFlow’s evolving threat landscape?*

**Comparative Study of Existing Models**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Model & Year | Core Technique | Data Set(s) | Reported Strengths | Key Limitations | LUFlow Perf.\* |
| 1 | Random Forest (2024)[[5]](#fn5) | Bagging of decision trees | LUFlow | High accuracy; interpretable | Memory-heavy; offline-batch | 91% accuracy |
| 2 | B-DRF Ensemble (2024)[[7]](#fn7) | Bootstrap + randomisation | LUFlow | 99% accuracy; lower variance | Batch-oriented; opaque | 99% accuracy |
| 3 | XGBoost (2024)[[8]](#fn8) | Gradient boosted trees | Multiple IDS sets | Handles imbalance; top F1 (99.4%) | Hyper-parameter heavy; CPU cost | 98–99%† |
| 4 | LightGBM (2024)[[9]](#fn9) | Histogram-based boosting | CIC-IDS17, UNSW-NB15 | Fast on large, sparse data | Weak on rare classes without tuning | N/A |
| 5 | SMO-ANN (2024)[[10]](#fn10) | Spider-Monkey optimised MLP | LUFlow + CIC-IDS17 | 100% binary accuracy | GPU needed; overfitting risk | 100% (binary) |
| 6 | CNN/LSTM Hybrid (2024)[[11]](#fn11) | Deep sequence learning | NSL-KDD, UNSW-NB15 | Learns complex patterns | High compute & RAM | N/A |
| 7 | Isolation Forest (2025)[[12]](#fn12) | Unsupervised anomaly det. | LUFlow | No labels required | 2.9% recall; high FP | 42.9% accuracy |

\*Published or reproduced. †Cross-validated on April-2025 LUFlow dump.

**Observations**

* Tree ensembles (Random Forest, XGBoost, LightGBM) dominate LUFlow thanks to its tabular, low-dimensional feature set but trade memory for speed.[[7]](#fn7)[[8]](#fn8)[[5]](#fn5)
* Deep models reach near-perfect accuracy on easier corpora yet are infeasible on typical edge gateways.[[10]](#fn10)[[11]](#fn11)[[13]](#fn13)
* Unsupervised methods remain attractive for unseen attacks but underperform badly on LUFlow’s imbalanced classes.[[12]](#fn12)
* None of the published studies report **sub-5 ms per-flow latency** or continuous deployment pipelines, leaving a gap between research prototypes and operational needs.[[13]](#fn13)[[14]](#fn14)[[15]](#fn15)

**Objectives & Justification**

**Objective 1 – Multi-Model Benchmarking & Selection**

*Build and evaluate a suite of classifiers—Random Forest (baseline), XGBoost, LightGBM and a lightweight DNN—on the latest LUFlow release to identify the model that maintains ≥ 90% detection accuracy while minimising inference time and memory footprint.*

**Rationale**  
Existing LUFlow studies focus on single algorithms in offline settings, hindering apples-to-apples comparison. A systematic benchmark will reveal the true **accuracy-latency-memory trade-offs** and inform which model can satisfy edge constraints where CPU, RAM and power are scarce. Feature selection, tree pruning and quantisation will be explored to squeeze additional efficiency from ensemble methods without significant accuracy loss.[[14]](#fn14)[[15]](#fn15)[[16]](#fn16)[[5]](#fn5)[[7]](#fn7)[[13]](#fn13)

**Objective 2 – Executable-Grade Deployment & Adaptation**

*Package the chosen model into the provided Joy-based streaming pipeline and compile it into a standalone executable, aiming to achieve ≤ 5 ms flow latency on Raspberry Pi-class hardware; and incorporate incremental–learning hooks to lift zero-day detection by ≥ 15% over the static baseline if possible.*

**Rationale**  
This approach avoids the overhead of running full Python environments, enabling practical deployment in real-world edge environments with limited CPU, memory, and power resources. By embedding the model directly into the flow-streaming pipeline, real-time detection can be performed efficiently and reliably at the network edge, crucial for timely threat mitigation in time-sensitive industrial and IoT applications.[[21]](#fn21)[[22]](#fn22)

**Alignment with Problem Analysis**

|  |  |
| --- | --- |
| Requirement | How Each Objective Meets It |
| **Resource constraint mitigation** | Obj 1 aims to quantify and minimise RAM/CPU.  Obj 2 aims to deliver a single-file executable and sub-5 ms inference budget, suitable for IoT gateways[[13]](#fn13)[[15]](#fn15). |
| **Emerging-threat adaptation** | Obj 2 aims to embed incremental learning to sustain performance on LUFlow’s evolving ground truth, closing the gap left by static classifiers[[18]](#fn18)[[19]](#fn19). (future scope) |
| **Real-world deployment focus** | The Joy + Executable toolchain aims to convert research code into an on-prem edge service with minimal dependencies[[17]](#fn17). |
| **Dataset alignment** | Both objectives exploit LUFlow’s 16-feature schema and weekly refresh cycle to ensure realistic evaluation and continuous validation[[5]](#fn5)[[6]](#fn6). |

**Standard Data Set Reference**

**LUFlow Network Intrusion Detection Data Set** – 16 engineered features, labels *benign/outlier/malicious*, weekly updates; collected via honeypots + CTI correlation, licence CC BY-SA 4.0.[[6]](#fn6)[[5]](#fn5)

**Future Scope**

In future work, adding an online concept-drift adaptation module—such as sliding-window partial-fit for tree ensembles or dynamic ensemble weighting—can help counteract performance decay as LUFlow’s weekly updates introduce novel attack patterns. This capability would allow the system to adapt continuously to evolving threats without requiring full retraining, thereby sustaining detection accuracy over time. Combining high operational speed with adaptive learning will make the intrusion detection system truly real-time and future-proof, meeting industry demands for security solutions in resource-constrained environments.[[17]](#fn17)[[18]](#fn18)[[19]](#fn19)[[20]](#fn20)

1. <https://www.darktrace.com/cyber-ai-glossary/real-time-threat-detection>

1. <https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/living-off-the-land-attack/>

1. <https://www.sciencedirect.com/science/article/pii/S0167404817301165>

1. <http://muhammetbaykara.com/wp-content/uploads/2018/10/ymtgunduz_intrusion.pdf>

1. <https://ieeexplore.ieee.org/document/10544158/>

1. <https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set>

1. <https://ieeexplore.ieee.org/document/10690014/>

1. <https://dl.acm.org/doi/10.1145/3702879.3702942>

1. <https://link.springer.com/article/10.1007/s42979-024-03369-0>

1. <https://www.nature.com/articles/s41598-024-68342-6>

1. <https://ieeexplore.ieee.org/document/10872423/>

1. <https://journal.uinjkt.ac.id/index.php/ti/article/view/44285>

1. <https://link.springer.com/article/10.1007/s43926-025-00099-4>

1. <https://arxiv.org/html/2403.01809v1>

1. <https://www.sciencedirect.com/science/article/abs/pii/S0045790622001859>

1. <https://link.springer.com/chapter/10.1007/978-981-97-1961-7_8>

1. <https://ieeexplore.ieee.org/document/10811056/>

1. <https://ieeexplore.ieee.org/document/10859433/>

1. <https://arxiv.org/abs/2203.05232>

1. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11598308/>

1. <https://appomni.com/what-is-threat-detection/>

1. <https://claroty.com/blog/boosting-resilience-critical-infrastructure-cyber-security>