Distributed Novelty Detection at the Edge for IoT Network Security

Luís Puhl, Guilherme Weigert Cassales, Helio Crestana Guardia, Hermes Senger

Luís Puhl

Universidade Federal de São Carlos, Brasil

luispuhl@gmail.com

September 13-16, 2021

Introduction

Context

- Growth of IoT devices and associated risks;
  - Heterogeneous devices;
  - Less frequent software updates;
  - Example: Mirai Botnet, infecting IP cameras and routers, generating 620 Gb/s [1].
- Network Intrusion Detection:
  - Detection by signature versus anomaly;
  - Fog and IoT network environment.

Proposal

- A system for IoT network intrusion detection implemented on the fog;
- The hypothesis of this work is: The MINAS algorithm can be run distributed in fog, reducing latency without classification quality reduction.

Introduction - Scenario

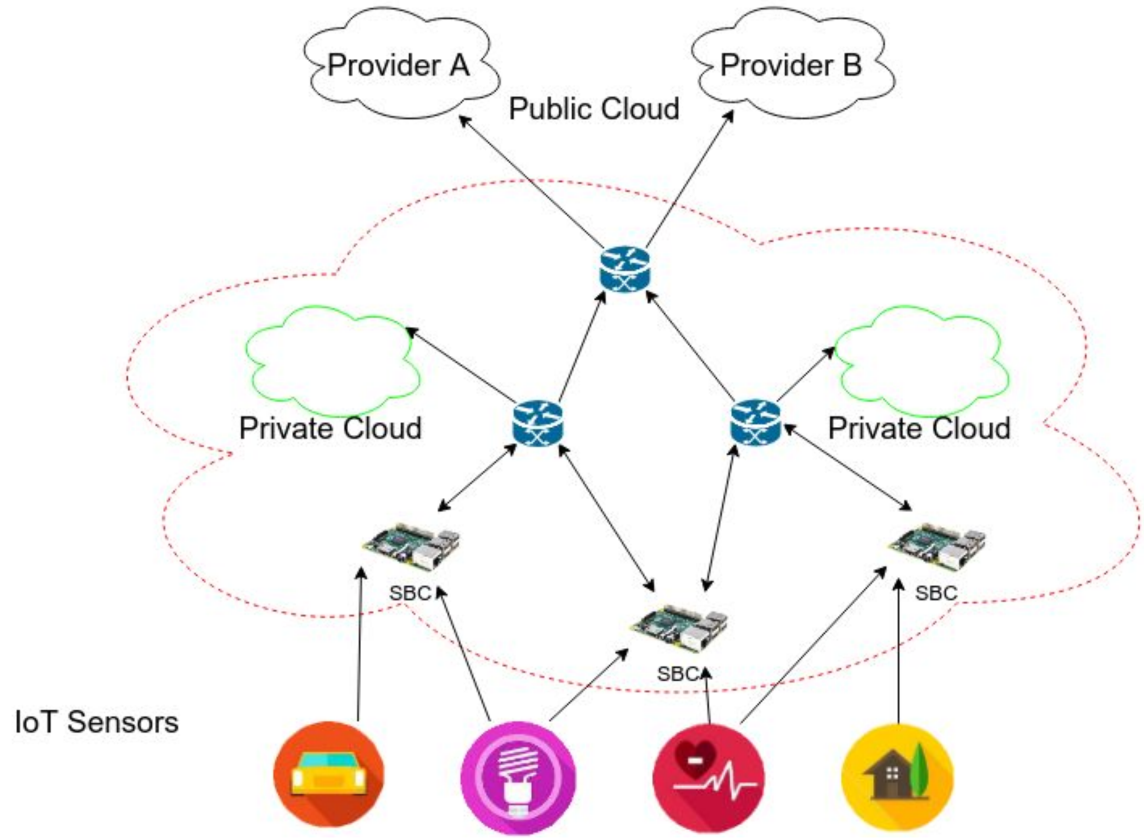

Fig. 1: IDSA-IoT [2] physical architecture and deployment scenario overview.

Related Work

BigFlow [5]:

- Intrusion via anomaly detection system capable of handling high speed networks;
- \+ Complete integration from flow descriptor extraction to alarms;
- \+ Capable of handling 10 Gbps with 40-core cluster;
- \- Weekly update with human specialist intervention;
- \- Cloud only.

Catraca [6]:

- Monitoring and threat detection system with stream computing and NVF;
- \+ Layered architecture allocated in cloud and fog;
- \+ Decision model based on decision tree;
- \- Flow descriptor extraction is done in fog, classification and detection on the cloud.

Related Work

IDSA-IoT Architecture [2]:

+ Evaluation of MINAS, ECSMiner and AnyNovel algorithms;
+ Task distribution on fog and cloud, focused on IoT;
- Implementation and evaluation in distributed scenario left open.

MINAS Algorithm [7]

- Analysis on space $\mathbb{R}^d$;
- Offline-Online learning;
- Classification in *known*, extension, and *novel* patterns or *unknown* labels;
- Decision model using spherical clusters and Euclidean distance;
- Clustering (K-means, CluStream) is used to find new patterns;
- Source available at http://www.facom.ufu.br/~elaine/MINAS.

Proposal

- MINAS as IoT NIDS on a fog environment - implements the IDSA-IoT architecture;
- Effects on classification quality due to distribution for scalability;
- Implement and evaluate viability and quality;

Method:

- Choice of technique and platform for implementation;
- Implementation of the IDSA-IoT architecture:
    - Extend fog usage to minimize latency;
- Experimentation with a suitable environment and dataset:
    - Classification quality metrics for validation;
    - Scalability metrics.

Implementation with MPI

- C, OpenMPI 4.0.4, compiled on Raspberry Pi;
- 2 modules: Root (single node) and Leaf (remainder nodes);
- Root: Sampler and Detector tasks;
- Leaf: Classifier (paralel) and Model Update tasks;
- Available at https://github.com/luis-puhl/minas-flink.

Implementation with MPI

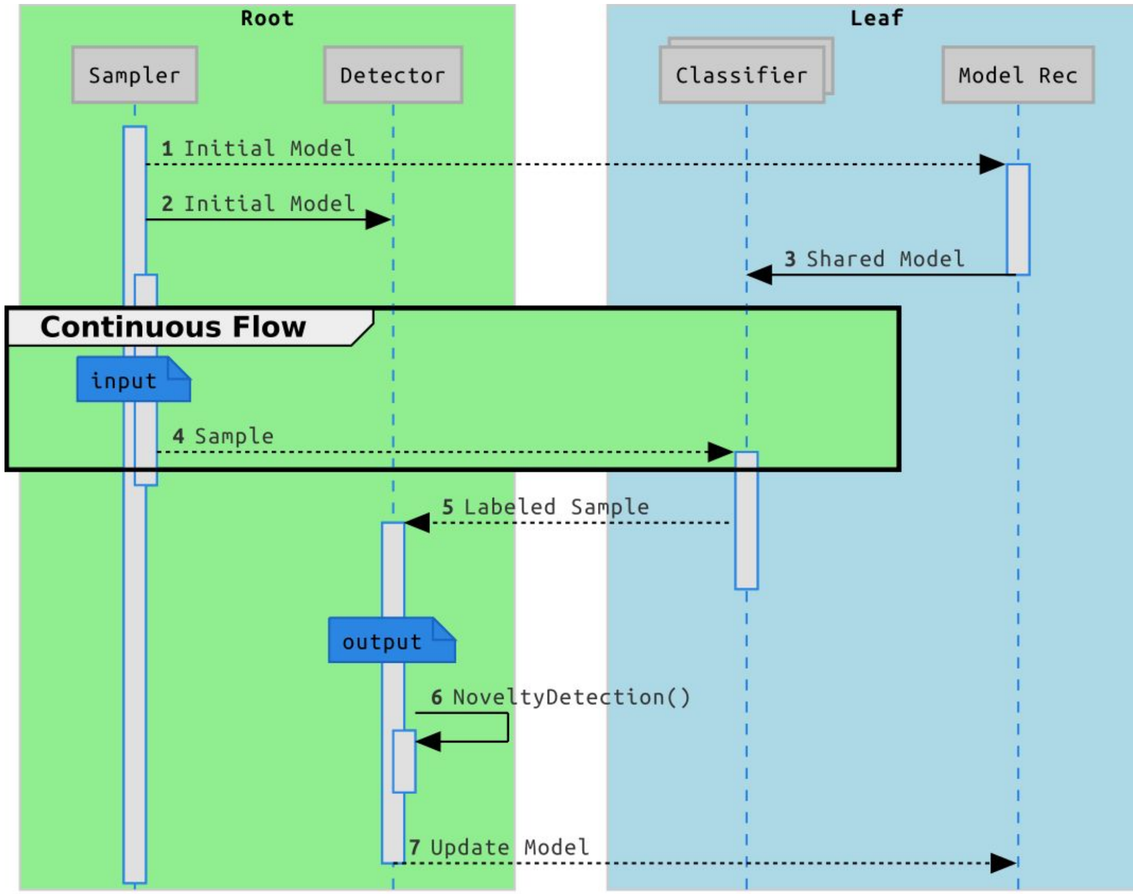

Fig. 2: MFOG life line overview.

Experiments and Results

Experimental Setup:

- Executed in a 3 Raspberry Pi 3B and Ethernet environment;
- December 2015 segment of Kyoto 2006+ data set [8]:
  - Available at http://www.takakura.com/Kyoto_data/.
  - 72 000 samples for training (offline) and 653 457 test (online) samples;
  - “N” (normal, 206 278 instances) known class;
  - “A” (attack, 447 179 instances) class to be detected as novelty.

Measurements:

- Multiclass confusion matrix with novelty label assignment;
  - Summary metrics (*Hits, Unknowns, Misses*);
  - Stream visualization of summary metrics;
- GNU Time: *Time*, *System* and *Elapsed* in seconds.

Experiments and Results

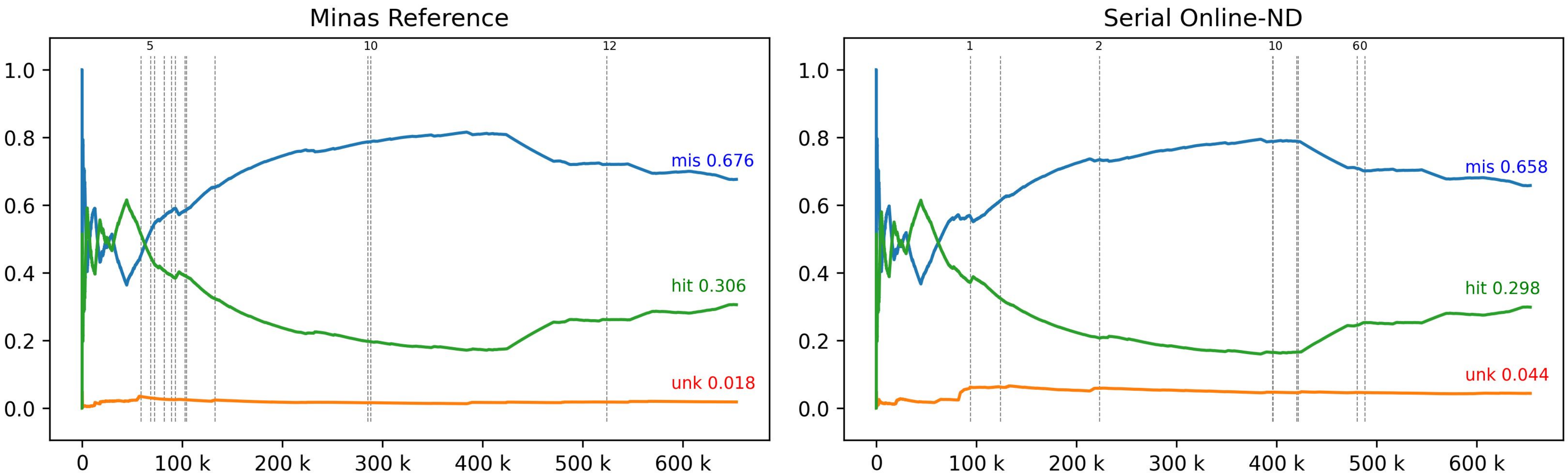

Fig. 3: Stream hits and novelties visualization.

Experiments and Results

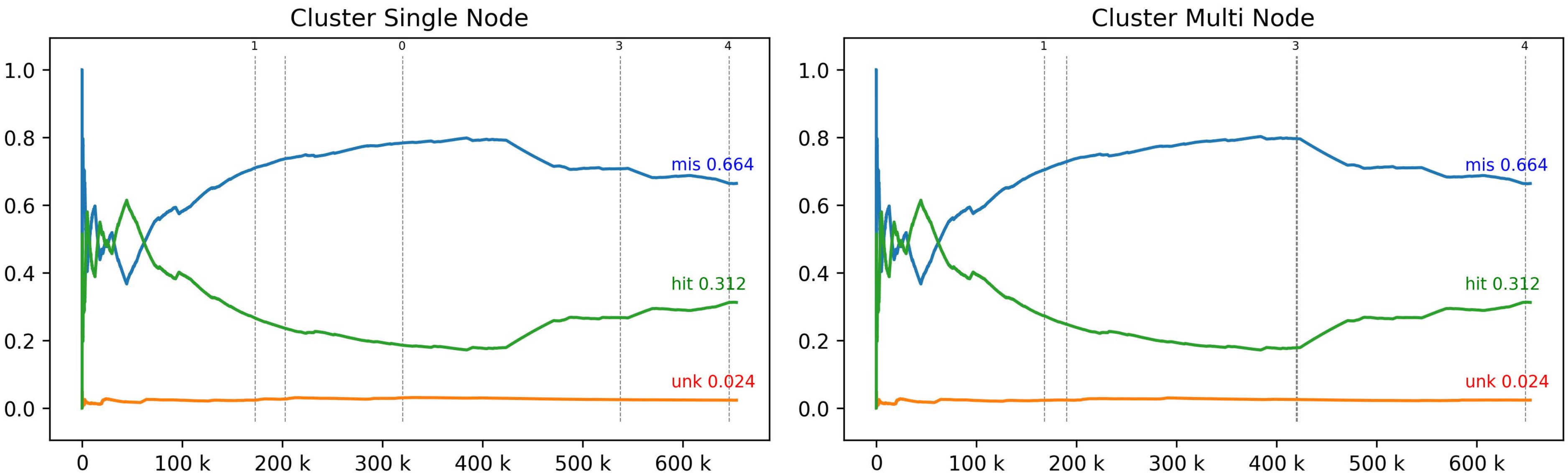

Fig. 3: Stream hits and novelties visualization.

Experiments and Results

Experiment Metric	*Ref* (a)	Offline	Sequential (b)	Single Node (c)	Multi Node (d)
unk	11980 0.018333		28567 0.043717	15370 0.023521	15499 0.023718
hit	199708 0.305618		195017 0.298438	204151 0.312416	204191 0.312478
err	441769 0.676049		429873 0.657843	433936 0.664061	433767 0.663802
Time (s)	2761.83	194.12	80.79	522.10	207.14
System (s)	7.15	0.075	11.51	47.77	157.61
Elapsed (s)	2772.07	194.27	93.03	145.04	95.38
Latency (s)	$4.24 \cdot 10^{-3}$		$1.42 \cdot 10^{-4}$	$2.22 \cdot 10^{-4}$	$1.46 \cdot 10^{-4}$
Processors	1	1	1	4	12
Speedup				0.6414092	0.9753617
Efficiency				0.1603523	0.0812801

Table 2: Collected Measures Summary.

Experiments and Results

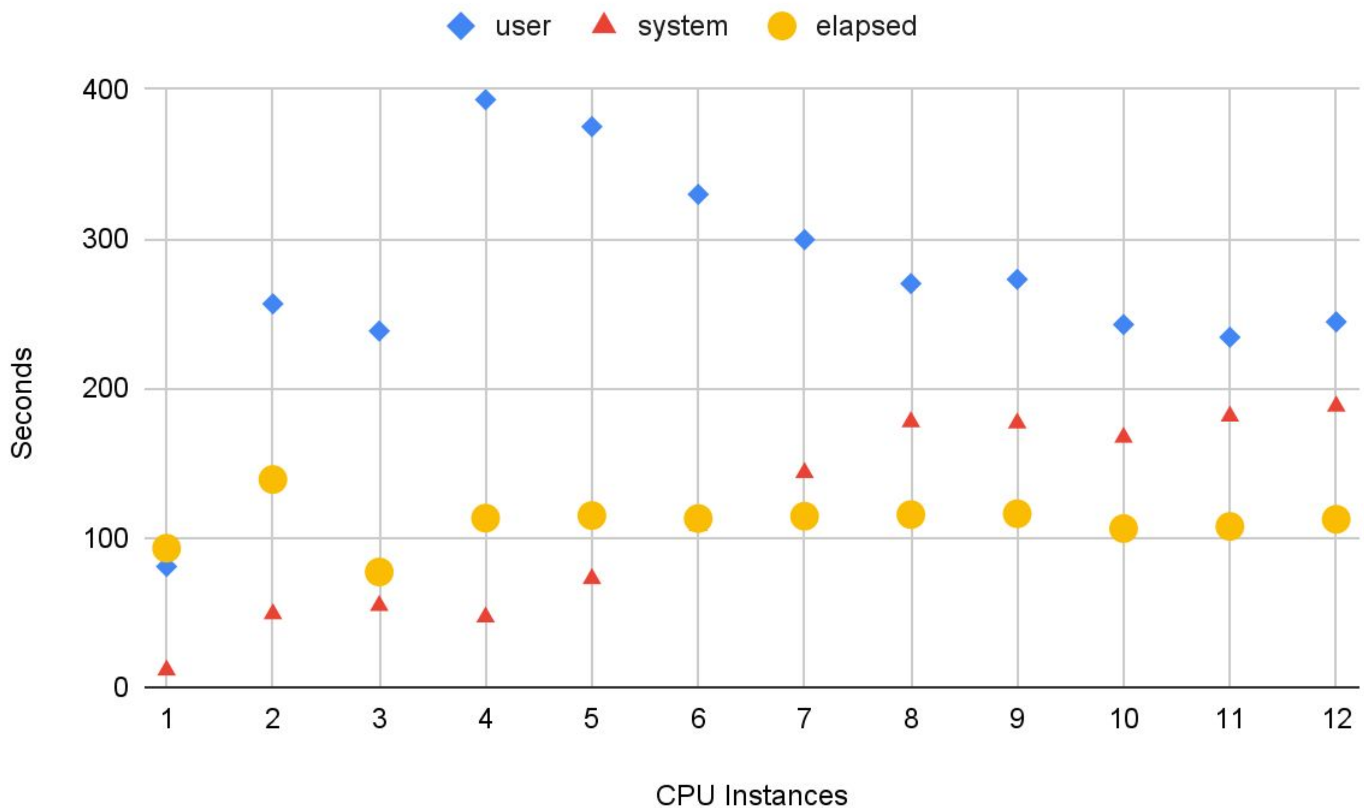

Fig. 4: Time measurements per added instance.

Conclusion

- Data Stream Novelty Detection as in Network Intrusion Detection or system behavior monitoring and analysis on IoT environments is still challenging due to data volume, latency and small devices constraints;
- Distributed data processing is a valid approach for novelty detection in this scenario;
- Our proposal, MFOG, a distributed architecture appling the Novelty Detection algorithm MINAS, was able to serve as an IoT Intrusion Detection system;
- Distribution causes an impact on the predictive metrics however, but a negligible loss of overall accuracy;
- The distributed model was faster than reference implementation, however parallel speedup was lower than expected.
- Further work:
  - Evaluate other Novelty Detection algorithms;
  - Change MINAS internal clustering;
  - Better load balancing strategies.