MUNI
ICS

Machine Learning in Intrusion Detection: An Operational Perspective

RNDr. Martin Husák, Ph.D.[1] **(husakm@ics.muni.cz)**
Darshan Manoj[2] **(dmanoj@usc.edu)**
Priyanka Kumar[3] **(kumar_p@utpb.edu)**

[1] Masaryk University, Czech Republic
[2] University of Southern California, CA, USA
[3] The University of Texas Permian Basin, TX, USA

October 28, 2024

Background

RDNr. Martin Husák, Ph.D.

- Researcher at Institute of Computer Science, Masaryk University, Czech Republic
- Member of Masaryk University's incident response team CSIRT-MU (`https://csirt.muni.cz/`)
- Recently also visiting researcher at The Cyber Center for Security and Analytics, The University of Texas at San Antonio, USA.

Paper background

- This research was inspired by discussions with machine learning experts (co-authors) and cybersecurity practitioners (namely members of CSIRT-MU)

Introduction

Motivation

- Machine learning is used everywhere, with promising results
- Intrusion detection is a prominent application of ML in cybersecurity research (ML-IDS for short)

Specific Problems

- There are thousands of research publications on ML-IDS...

  ... but only a very few open source prototypes!
- The topic is frequent in a start-up scene...

  ... but how many usable commercial products exist?

Surveying ML-IDS

Literature Survey

- Hundreds or thousands of results, depending on keywords (* *learning*) and library (Google Scholar, IEEExplore, ACM DL)
- Looking up survey papers does not help, either –
  – over 30 were published in 2023 only!

Typical ML-IDS paper

- Motivated by recent attacks, but uses obsolete datasets
- Input is dataset CSV, not PCAPs of network flows
- Claim accuracy of more than 99 %, binary classification only
- No implementation nor attempt to deploy in live traffic

Surveying ML-IDS

Commercial products

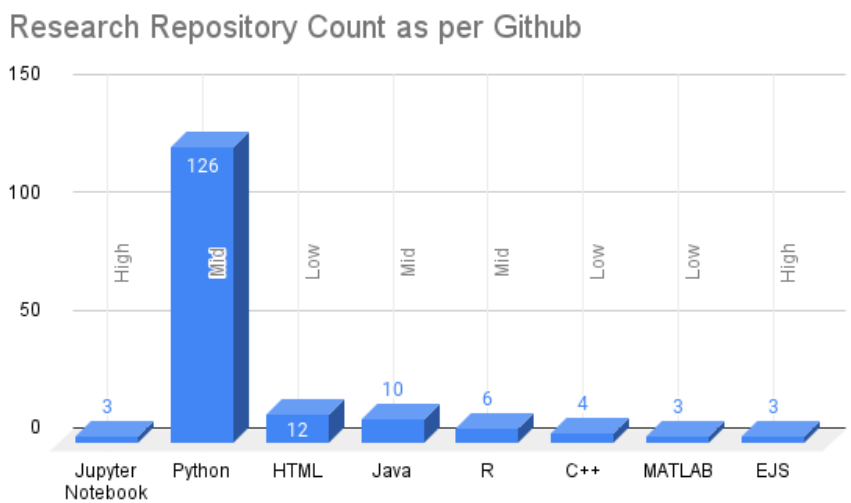
- Darktrace, ExtraHop, Vektra, ...
- Interesting, but unclear how they work, mixed reviews

Open-source example - StratosphereLinuxIPS

- `https://github.com/stratosphereips`
- Highly recommended work from operational perspective

GitHub survey

- Most of the other repositories are just simple experiments



Criticism of ML-based Intrusion Detection

- R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in 2010 IEEE Symposium on Security and Privacy, 2010.
- G. Apruzzese et al., “On the effectiveness of machine and deep learning for cyber security,” in 2018 10th International Conference on Cyber Conflict, 2018.
- D. Arp et al., “Dos and don’ts of machine learning in computer security,” in 31st USENIX Security Symposium, 2022.
- D. Arp et al., “Lessons learned on machine learning for computer security,” IEEE Security & Privacy, 2023.
- F. Ceschin et al., “Machine learning (in) security: A stream of problems,” Digital Threats: Research and Practice, 2023.
- G. Apruzzese, P. Laskov, and J. Schneider, “Sok: Pragmatic assessment of machine learning for network intrusion detection,” 2023.
- G. Apruzzese et al., “The role of machine learning in cybersecurity,” Digital Threats: Research and Practice, 2023.
- A. Corsini and S. J. Yang, “Are existing out-of-distribution techniques suitable for network intrusion detection?” 2023.

Issues of ML-based Intrusion Detection

What is the use case for ML-based IDS?

- Various environments call for various solutions
  - Huge amounts of data in backbone networks and clouds
  - Low computational capacity of IoT networks
  - Rigidity of OT vs. variability of generic IT
- Let's consider IDS in a campus/enterprise network
- Intrusion detection vs. anomaly detection

What data are on the input?

- Most often a dataset with CSV input is used
- How to obtain the feature vectors?
- Raw data –> input data is a challenging problem
- Raw packets/PCAPs x NetFlow/IPFIX and similar aggregates

Issues of ML-based Intrusion Detection

Are they data processed in batch or in a stream?

- Huge design issue when implementing an IDS
- Stream allows for faster detection, if feature set and method allows for it
- Batch is still relevant, e.g., in centralized analysis of data collected by distributed probes

What features to use?

- Existing datasets tens of different features
  - Not all of them are available in common formats like NetFlow/IPFIX (e.g., *ct_srv_src/dst* in UNSW-NB15)
  - Some cannot be obtained any more (encryption) or require collaboration with end hosts (e.g., *logged_in* in KDD'99)
- Creating a custom packet parser should be well justified
  - Compare to the time and effort spent on developing precise NetFlow probes

Issues of ML-based Intrusion Detection

Which model (and optimizations) to choose?

- According to related work, there are multiple options achieving accuracy and precision over 99 %
- The final selection will depend on other factors than accuracy
  - computational performance, easy of use, configurability, ...

How to train the model and how transferable it is?

- Ideally a combination of datasets and background traffic
  - Training on dataset only will produce FPs in operations
  - Custom data will lack ground truth
- Potential solution – Siamese neural networks[1]

[1] M. Pawlicki, R. Kozik, and M. Chora´s, "Improving siamese neural networks with border extraction sampling for the use in real-time network intrusion detection," in 2023 International Joint Conference on Neural Networks (IJCNN), 2023.

Issues of ML-based Intrusion Detection

How often to retrain the model and who should do it?

- Conceptual drift – a trained model will lose accuracy over time
- Re-training the models is recommended, but how often?
- Is an average IDS operator able to re-train the model correctly?
- Can we offload the work to experts?

What is the computational performance of ML-IDS?

- Unaddressed question in most of the works
- If addressed, then either only training time or time to process the whole dataset (in batch) in seconds
- Training time is usually less important (not executed often)
- Imagine a stream-based ML-IDS, what is its throughput in Gbps?

Issues of ML-based Intrusion Detection

How are the alerts raised, how do they look like, and how many are there?

- Unaddressed in related works, which ends at classification
- Extreme risk of information overload for the operator
- For example, SLIPS implements a plethora of heuristics and thresholds to trigger optimal amount of alerts

What options does a user have to configure or modify the IDS?

- Re-training the whole model is impractical
- Advantage of ensemble-based solutions (plug-in models)
- Workflow automation – not all alerts should trigger an automated response

Discussion

Is it really better than traditional IDS?

- In terms of accuracy and precision - yes, but other aspects?
- Beware of trade-offs – aren't we losing key features?

How would a good ML-IDS look like?

- Takes NetFlow or other standard as an input
- Capable of processing large volumes of data
- Distinguishes between different types of attacks (e.g., multi-label classification or ensemble of specialized models)
- Pre-trained models provided by vendors or community?
- Smooth blending of pre-trained models and background traffic
- Explainable AI is not exactly what we need

Discussion

Is it really worth it?

- *Pet and Cattle* analogy in DevOps
- Are you willing to spend time and effort to make it work and oversee it all the time? ML-IDS is your pet
- Can you deploy it all over your network in the blink of an eye and let it run with minimal interventions? ML-IDS is not the cattle yet

Are there any alternative approaches?

- Directly applying ML is not the only way
- AI-assited generation of IDS signatures nad rules[2]
- Potential for faster adoption and persuasion of users into ML

[2] M. Zipperle, Y. Zhang, E. Chang, and T. Dillon, "PARGMF: A provenance-enabled automated rule generation and matching framework with multi-level attack description model," Journal of Information Security and Applications, 2024.

Conclusions

Conclusion

- Machine learning in intrusion detection is a hyped topic
- ML performance is not everything – actually, it is only a small part of any usable ML-IDS
- Researchers spent enormous efforts on optimizing only one aspect, there is a lot more to figure out and implement

Future work

- Proper surveys, field trials of available implementations, ...
- Steering the research and development community
- Doing the hard work of actually implementing ML-IDS :)

MASARYK
UNIVERSITY