GPU-based Classification for Wireless Intrusion Detection

Alina Lazar[1], Alex Sim[2], Kesheng Wu[2]

[1]Department of Computer Science and Information Systems, Youngstown State University

[2]Scientific Data Management Research Group, Lawrence Berkeley National Laboratory

Introduction and Motivation

- 5G wireless technologies and IoT grow in size and complexity
- Robust network security systems, such as intrusions detection systems (IDS) become important
- Passive wireless traffic monitoring tools collect huge amounts of data
- RAPIDS.ai cuML library on Graphics Processing Units (GPUs) can speed-up the training of machine learning models

5G and IOT

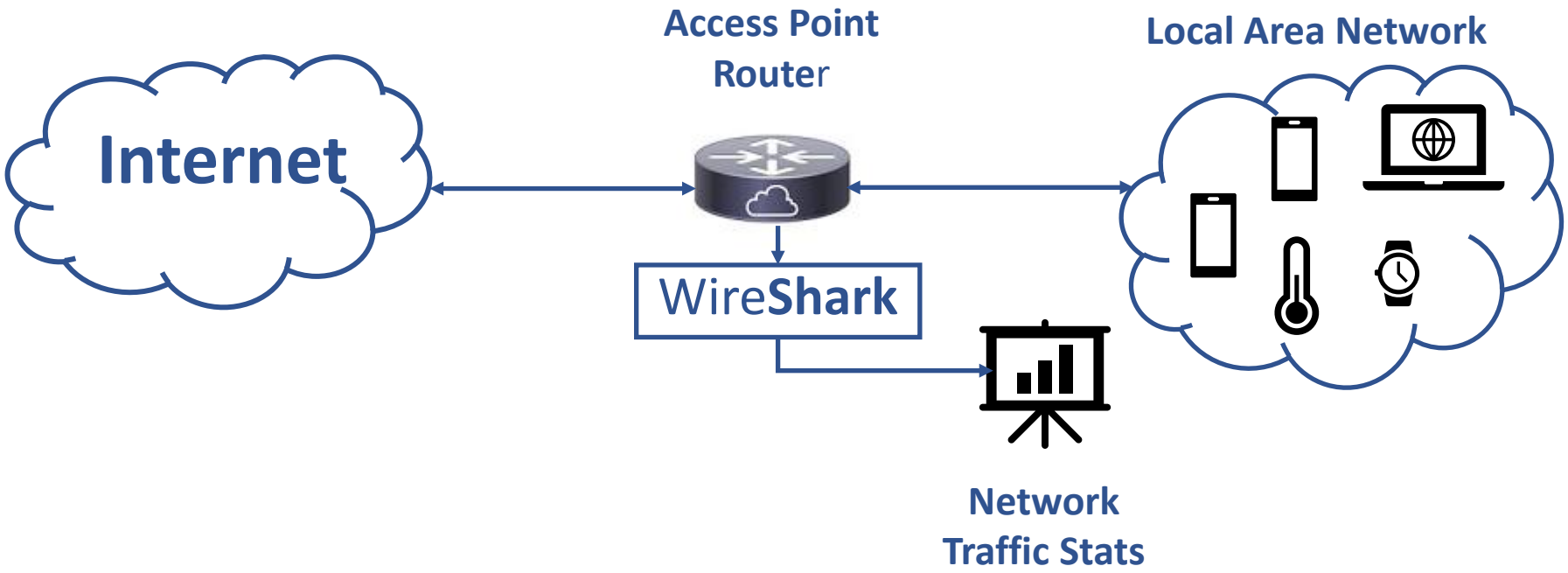
- Sensors and wireless devices interconnected through the Internet are becoming extremely important
- Cyberattacks are threatening banking, online shopping, e-health and other digital services

Wireless Network Intrusion

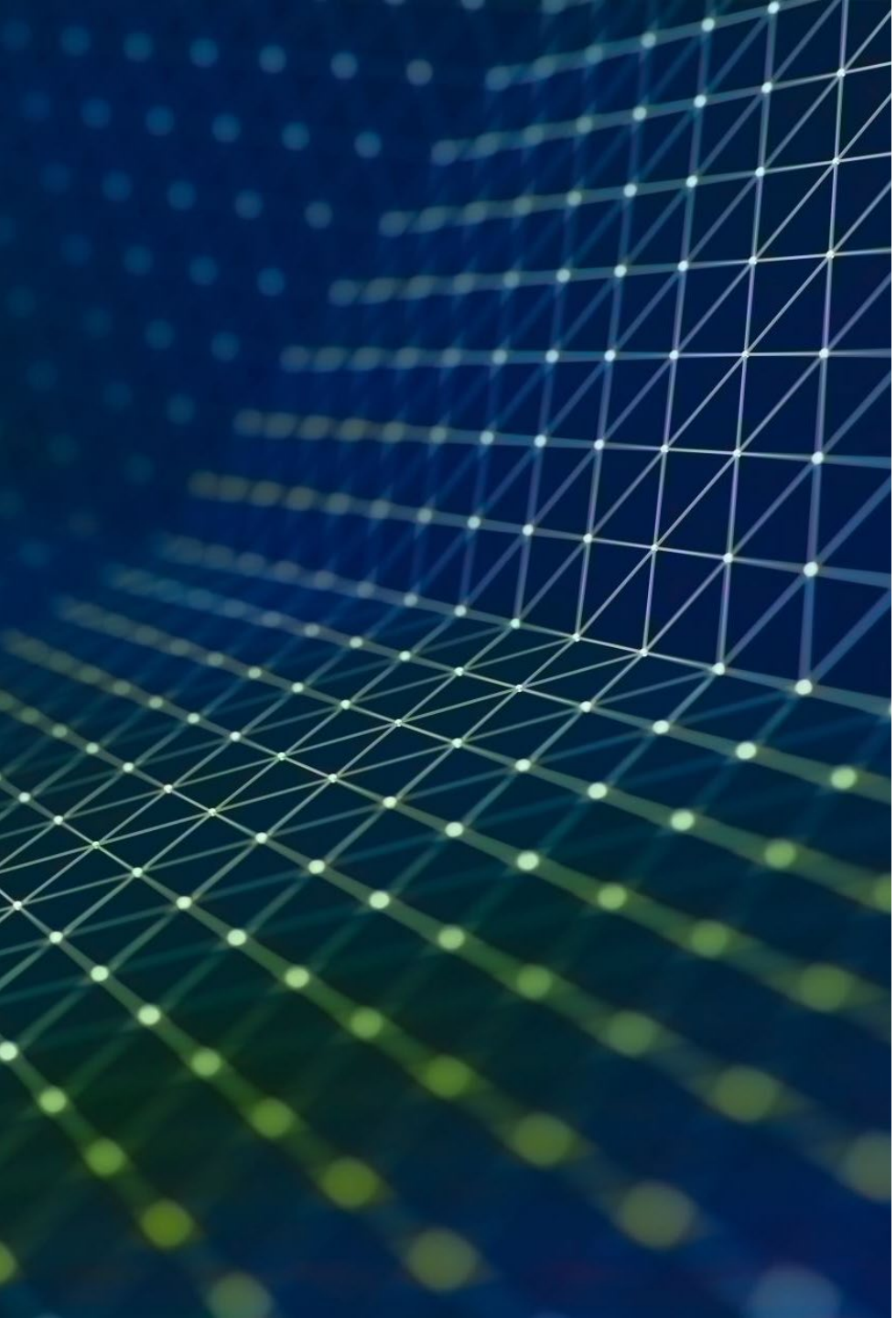
- The 802.11 protocol is commonly used to implement WiFi networks.
- Wireless local area networks connect not only computers and cell phone, but also personal devices and IoTs.
- Security based on WEP and WPA/WPA2 protocols.

- New penetration tools make it easier to automate network attacks.
- Wireless networks are more vulnerable compared to wired networks since they are open.
- Depending on the security protocol many types of attacks are possible.

AWID Simulated Datasets



- Research Question: **Using large wireless network datasets can we identify minimal sets of features that correctly discriminate between the "normal" versus "attack" transfers?**
- Typical categories of network attacks include **Impersonation, Flooding and Injection**



Current Drawbacks

Problems:

- Not real-time, only checked when something is wrong
- Large datasets to analyze

Ideal case:

- Automate the detection of wireless intrusions and raise alerts
- Wireless network traffic data is high volume, heavy stream of high dimensionality data
- Few training (labeled) datasets available

Main Contributions

Supervised Machine Learning:

- Classification experiments performed on the AWID wireless network data using several RAPIDS.ai classification methods.

GPU Training and Inference:

- RAPIDS.ai provides efficient implementations of classification algorithms that run on NVIDIA GPUs.

Training Speed-up:

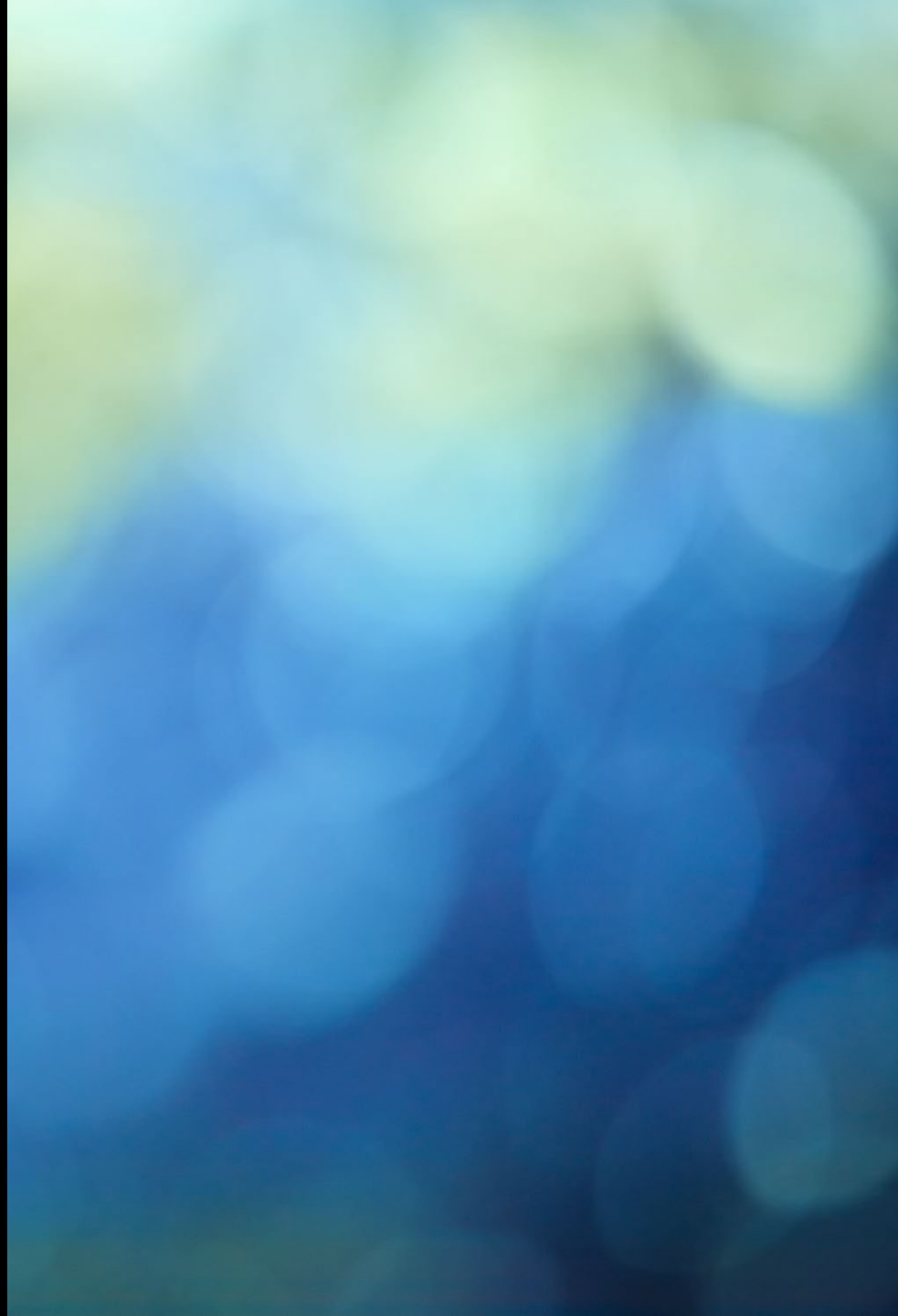
- Using the RAPIDS AI implementations, we are able to train classification algorithms over large intrusion detection datasets up to 65x faster compared to conventional CPU versions.

Previous Work

In 2015, Kolias collected a set of wireless intrusion detection datasets and made them publicly available for research.

Random Forest (RF) models are good choice to implement NIDS because of their ability to overcome overfitting and to perform well on imbalanced datasets, with missing values and with large number of attributes.

The XGBoost algorithm was used by Bhattacharya et al. on a reduced dataset for classification. The method was demonstrated on a dataset with 43 features and 125,973 instances and ran on Google Colab using GPUs, but no training times or comparison with the CPU timings were reported.



Random Forest

An overview survey of Random Forest (RF) applications for IDSs was presented by Resende.

RF deal well with imbalanced datasets, large number of features and categorical features as well as numerical features.

An advantage of RF is that models can be trained in shorter amount of time compared to deep learning.

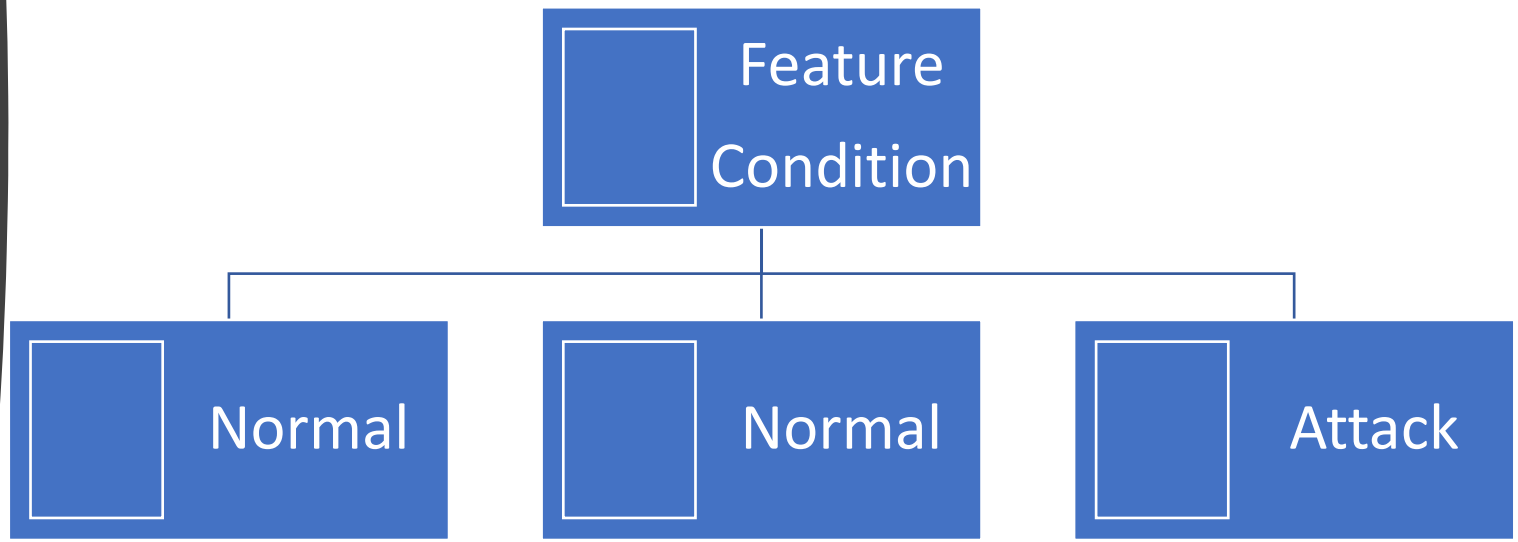
Another advantage is the ability to easily perform feature ranking and selection.

Tree-based Methods

Random Forest

XGBoost

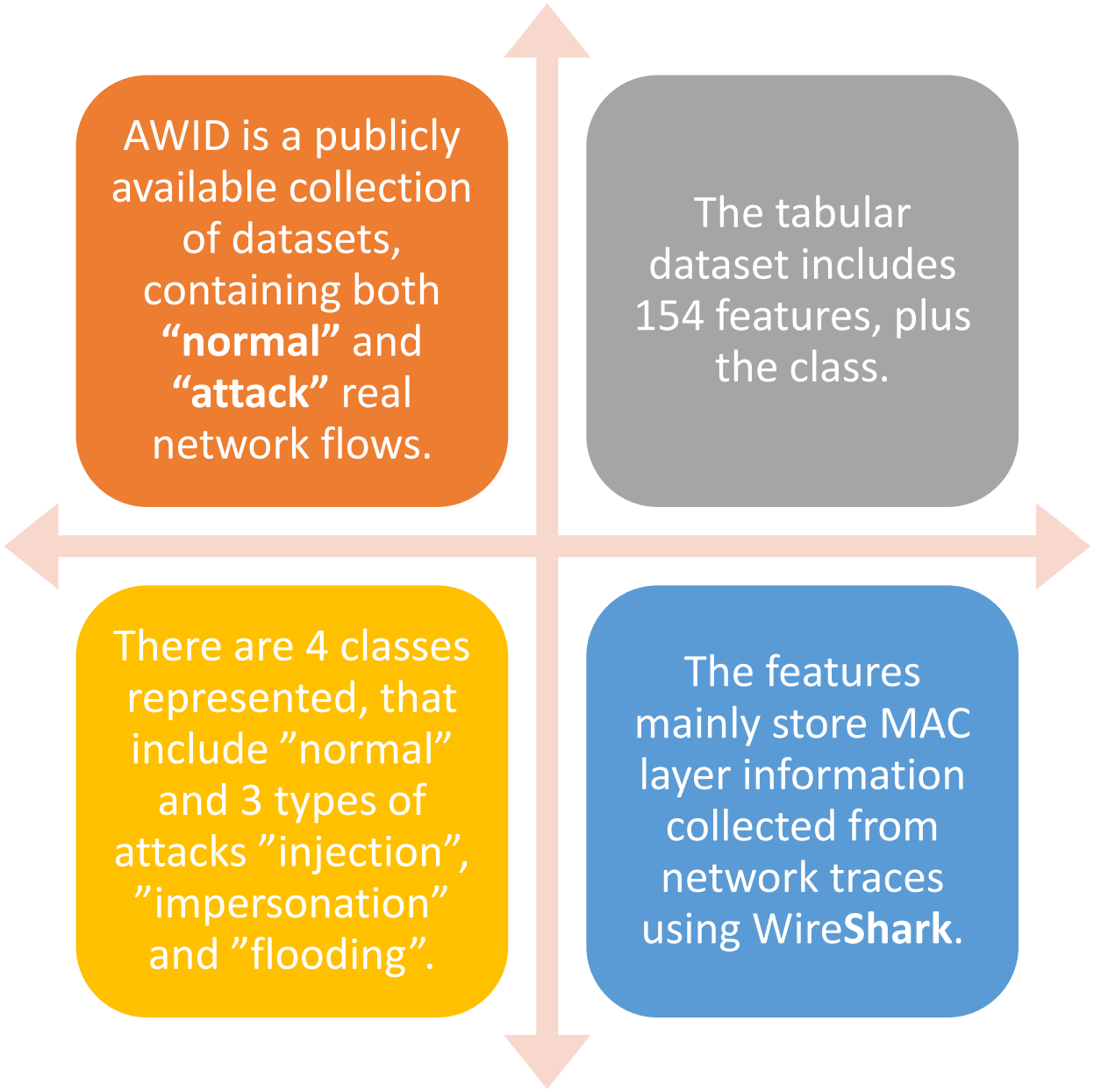
CatBoost



Algorithm Complexity

	Training	Prediction
Naive Bayes	$O(np)$	$O(p)$
KNN	-	$O(np)$
Linear SVM	$O(p^2n + n^3)$	$O(n_{sv}p)$
RF	$O(n^2pn_{trees})$	$O(pn_{trees})$
Gradient Boost	$O(npn_{trees})$	$O(pn_{trees})$

AWID (The Aegean WiFi Intrusion Dataset)



	Normal	Injection	Impers.	Flooding
Large AWID	8,336,139	84,741	461,707	10,134

AWID (The Aegean WiFi Intrusion Dataset)

- The data is already divided into two datasets on called training (AWID-CLS-F-Trn) and one called testing (AWID-CLS-F-Tst).
- The ratio between the number of **”normal”** and **”attack”** instances is 14:1
- This shows the class imbalance, that occurs when classes are not equally represented in the dataset.

Data Preprocessing

- The AWID datasets contain multiple features with different data types and value ranges.
- There is only one string feature, namely SSID, and all the other ones have numeric or nominal values.
- Features that represent MAC addresses are stored as hexadecimal values and need to be converted before the analysis.
- Particularly, in the training dataset there are many missing values and after converting these to zeros, many features have more than 99% zero values. After removing the mostly "zero" features, 100 features were left.
- A typical MAC address takes values in the $[-2^{31}, 2^{31} - 1]$, the typical value of subtypes (feature wlan.fc.subtype) is an integer between 0 and 12.
- All features are normalized using the MinMax Scaling procedure.

Experiments and Results

- The experiments were performed on computing nodes at the Ohio Supercomputer Center.
- These computing nodes have 48 CPU cores, 384 GB of memory, and 2 NVIDIA Volta V100 GPUs.
- We used a RAPIDS 0.18 conda environment including the cuDF and cuML libraries together with NVIDIA libraries.
- Multi-class classification using Naïve Bayes, KNN, SVC, Random Forest, XGBoost, and CatBoost is performed.

Experiments and Results

- We ran experiments in a loop based on the training dataset size.
- Inside the loop:
  - the data is divided into training and testing
  - scaling is applied to both training and testing
  - the model is trained
  - prediction is done on the testing dataset to evaluate the model and get the accuracy.

Accuracy for the CPU vs GPU Reduced Feature Sets

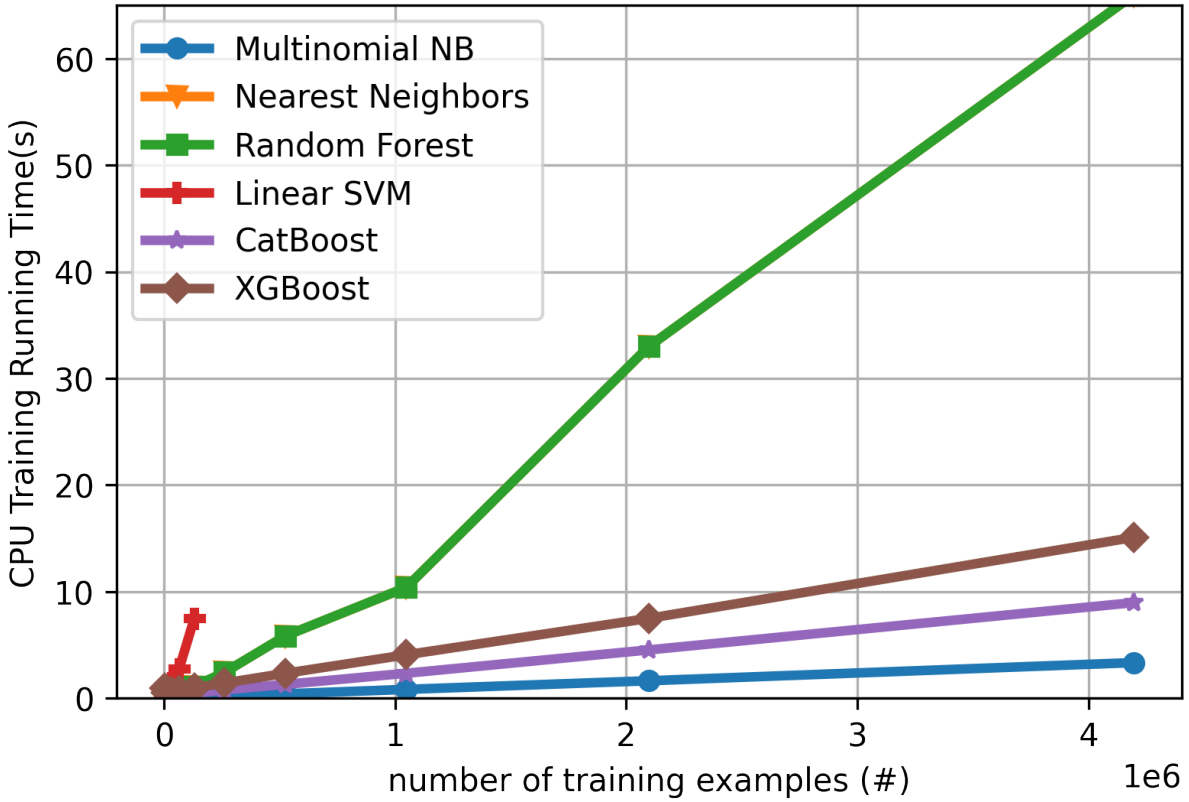
	Acc.	Prec.	Recall	F1
MultinomialNB	53.67	41.29	26.67	20.87
KNN	-	-	-	-
SVC	100	100	100	100
RF	95.70	35.72	48.9	39.45
XGBoost	100	100	100	100
CatBoost	100	100	100	100

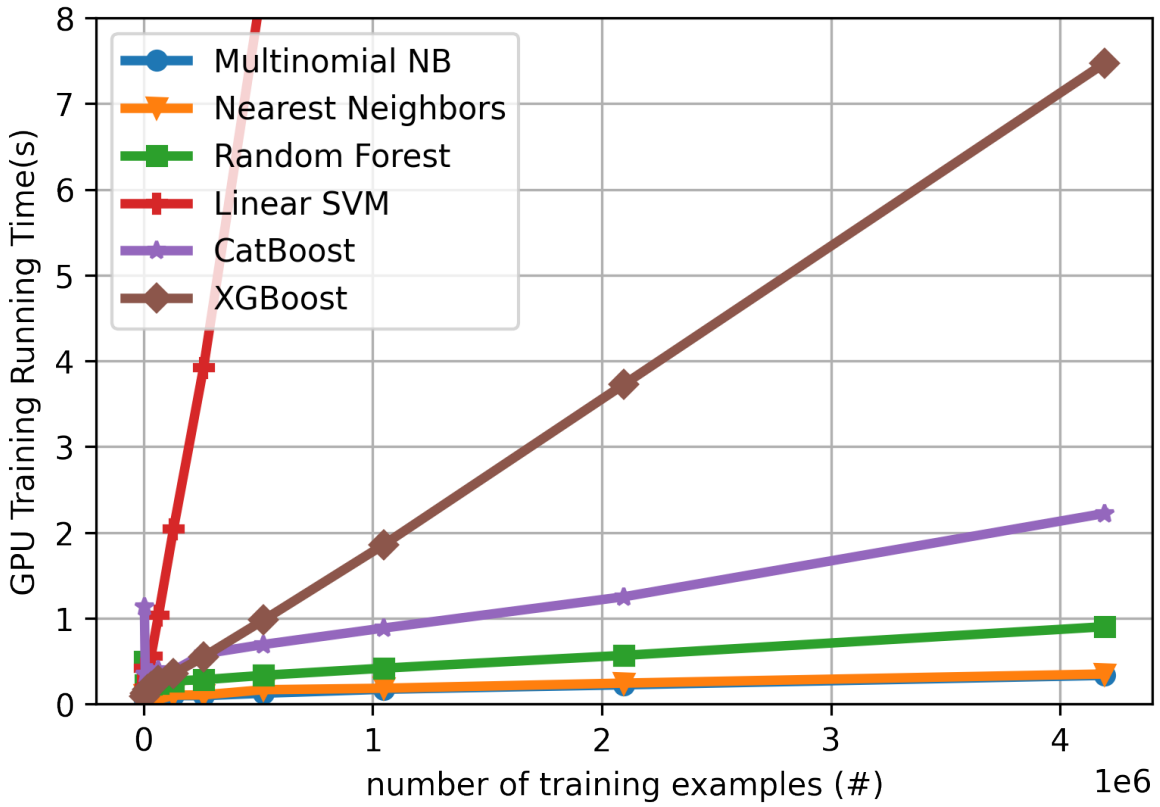
(a) CPU

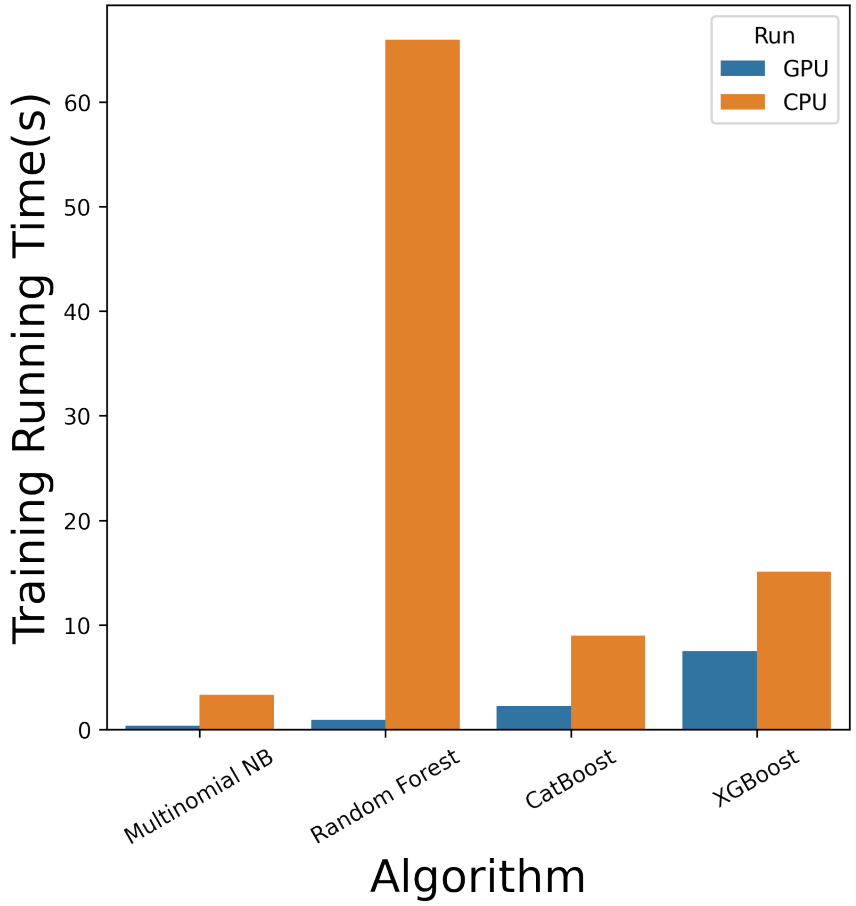
	Acc.	Prec.	Recall	F1
MultinomialNB	69.72	45.51	29.15	24.98
KNN	93.38	27.12	48.46	28.13
SVC	100	100	100	100
RF	97.51	42.92	49.30	45.53
XGBoost	100	100	100	100
CatBoost	100	100	100	100

(b) GPU

Training Running Time Comparison on CPU and GPU







Training running time comparison on CPU and GPU

- The largest performance gap is seen for Random Forest, where the GPU version improves by 65x.
- For the other algorithms, the improvement varies between 2x and 5x.

Conclusions

This paper presents a scalable machine learning workflow to speed-up network intrusion detection and attacks classification over large 5G datasets. The results show a speedupup to 65-fold on GPUs.

- Uses the RAPIDS.ai cuML library and the CatBoost library to compare these implementations with classical scikit-learn CPU implementations.
- This pipeline can be adapted to other intrusion detection tasks processing and interpretation tasks, aiming to provide efficient and scalable solutions to many applications.

Conclusions

The proposed pipeline may be adapted to other intrusion detection datasets to provide efficient and scalable solutions for these important applications.

In future, we plan to extend the current system:

- To use multiple GPUs to extend the presented methods to the full AWID training and testing datasets.
- To apply the same approach to other datasets to investigate the generalization capabilities of the presented method.

Acknowledgements

This work was supported by:

- *The Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.*
- *This research used resources of the Ohio Supercomputer Center and the National Energy Research Scientific Computing Center.*

Questions

- Alina Lazar alazar@ysu.edu
- Alex Sim asim@lbl.gov
- Kesheng Wu kwu@lbl.gov