Project overview :

The main purpose of this project is to develop an AI powered threat detection with LLM explanaition and real-time monitoring. The solution should be scalable and deployable in different environments.

Development solution :

The datasets used (with their correspondant 3 models) are from

3 AI models,each one is trained on a different dataset (IoT Malicious Detection Dataset - https://www.kaggle.com/datasets/agungpambudi/network-malware-detection-connection-analysis ,

Malware Detection in Network Traffic Data - https://www.kaggle.com/datasets/agungpambudi/network-malware-detection-connection-analysis

And Network Traffic Data-Malicious Activity Detection https://www.kaggle.com/datasets/advaitnmenon/network-traffic-data-malicious-activity-detection)

The choice of these 3 different datasets is to create 3 models, each one trained on specific field of data, although they have a lot of common columns, the goal is to combine them together into our pipeline to have 3 models each one trained on a different domain of packets for a better prediction/ packets classification together.

Prerequisites used to start developing :

2 kaggle notebooks already publically available, we didn't invent the wheel from scratch to see what will work and what won't.

https://www.kaggle.com/code/rem4000/xgboost-iot-malicious-detection-99-99-accuracy

https://www.kaggle.com/code/istiakahammedeee/explainable-ai-techniques-for-intrusion-detection

We have gone through developed estimates / code already existing in kaggle , learnt what model will perform the best. Then we started coding our 3 notebooks each one related to a specific dataset.

We have transformed the datasets to have the most common columns that can be used in real time packet capture and re built models based on that.

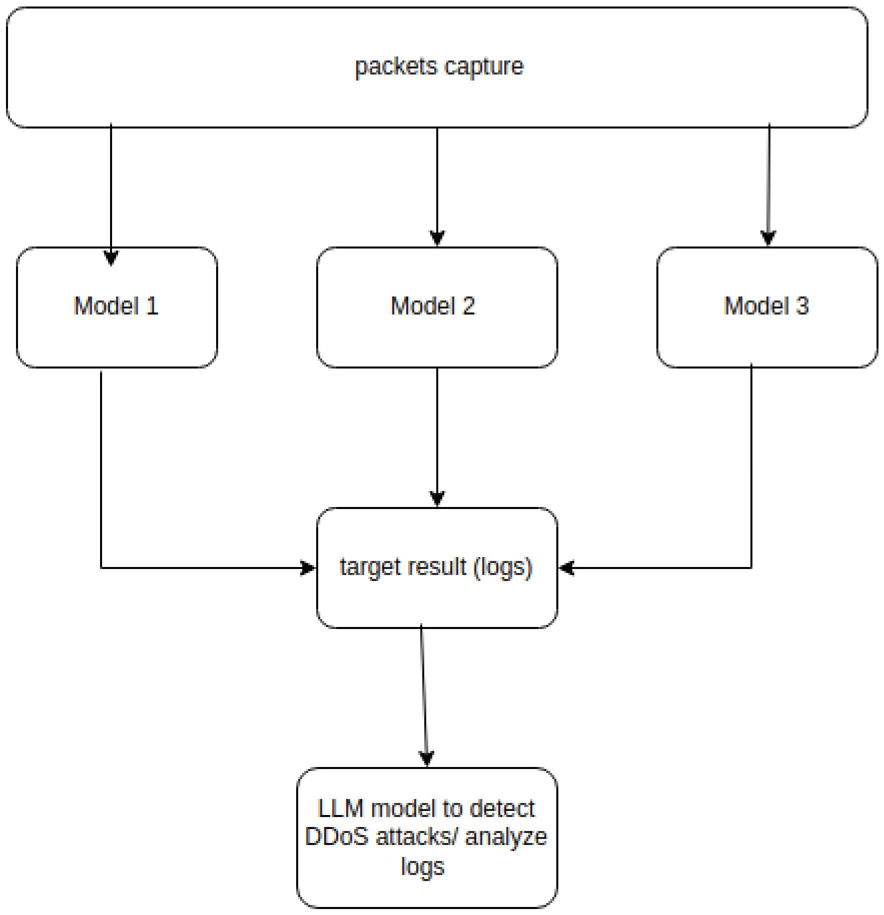
The XGboost model gave an accuracy of 99 to 100% with some small errors which can be found in confusion matrix.

The notebooks are in the git repository named after the python files on which we have saved the pikel file on) example : notebook detector1 py is the notebook which we used its model results in the detector1.py file..

But these notebooks data are a “sandbox” they aren’t real-time !

That's why we have continued the implementation to see the models execution and result on real-times scenario.

Setting up the project architecture :



Since the 3 models will be combined (works in parrallel) , they should have a unified input, which are common columns to extract from each of the 3 datasets.

The models testing / accuracy is based on its training on these common columns only.

The models will be saved and imported to the pipeline once trained and validated.

Each model will give a prediction to the given packet , the logs will be inserted and analyzed by an LLM model to detect any sort of redendency or a DDoS attack or threat.

Setting up environments and testing

After training and saving the models from the notebooks, we test the models on :

Development enivornment : running script locally and generate a flow of packets to capture.

Staging environment : running the script from a virtual machine and capture packets sent from kali linux VM.

Production env: simulating real deployment on azure by checking the results after deployment (run the LLM on a VM to filter the logs)

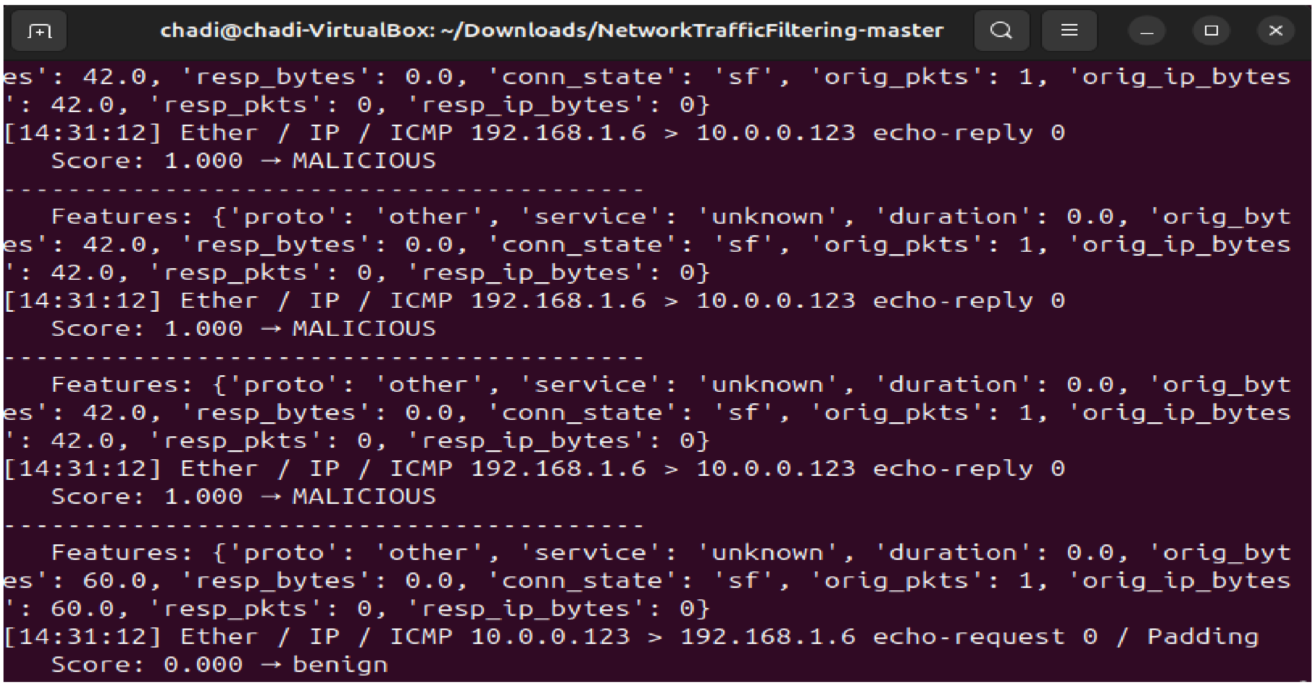
Steps :

After developing the 3 notebooks and saving the models,

Staging is to test locally between 2 VMs , simulating threat attacks with Kali linux

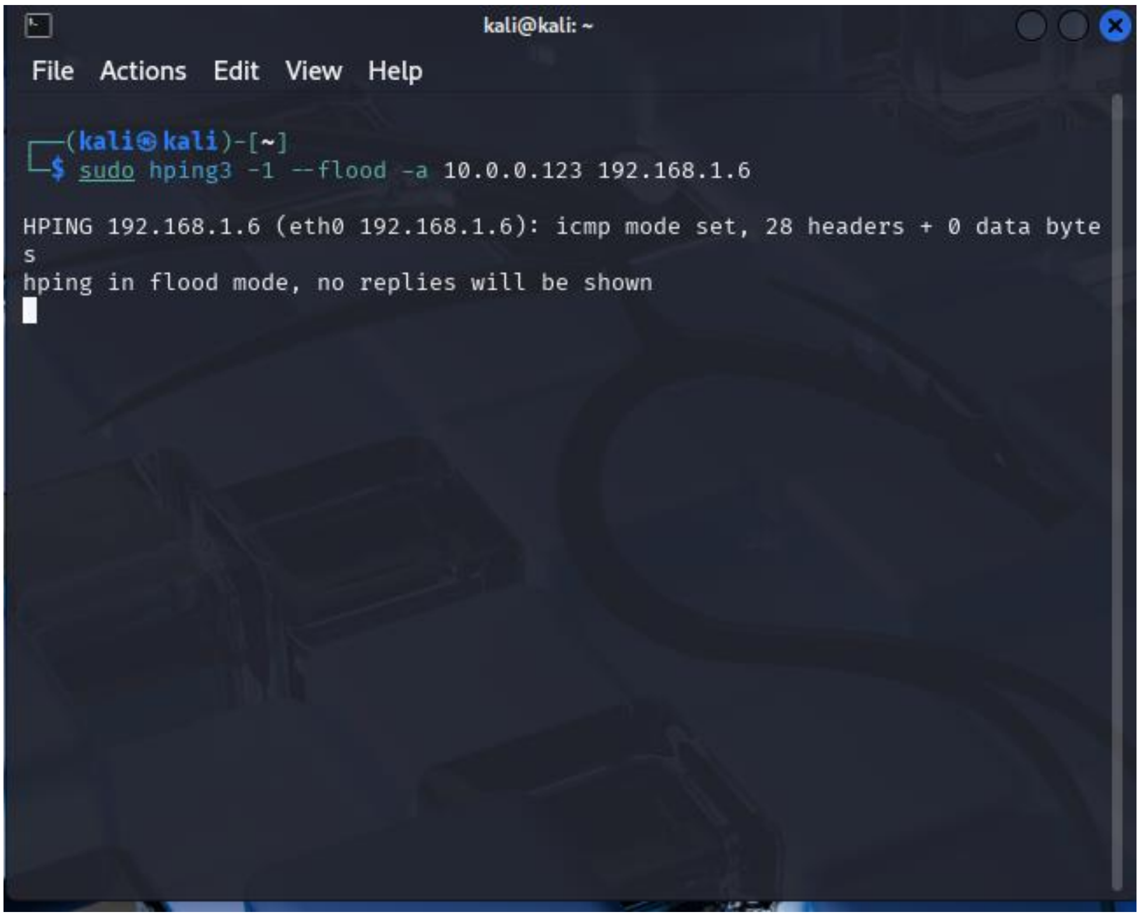
Example : ICMP flood. , detector = detector1.py

Model execution


chadi@chadi-VirtualBox: ~/Downloads/NetworkTrafficFiltering-master

```
es': 42.0, 'resp_bytes': 0.0, 'conn_state': 'sf', 'orig_pkts': 1, 'orig_ip_bytes
': 42.0, 'resp_pkts': 0, 'resp_ip_bytes': 0}
[14:31:12] Ether / IP / ICMP 192.168.1.6 > 10.0.0.123 echo-reply 0
   Score: 1.000 → MALICIOUS
----------------------------------------
   Features: {'proto': 'other', 'service': 'unknown', 'duration': 0.0, 'orig_byt
es': 42.0, 'resp_bytes': 0.0, 'conn_state': 'sf', 'orig_pkts': 1, 'orig_ip_bytes
': 42.0, 'resp_pkts': 0, 'resp_ip_bytes': 0}
[14:31:12] Ether / IP / ICMP 192.168.1.6 > 10.0.0.123 echo-reply 0
   Score: 1.000 → MALICIOUS
----------------------------------------
   Features: {'proto': 'other', 'service': 'unknown', 'duration': 0.0, 'orig_byt
es': 42.0, 'resp_bytes': 0.0, 'conn_state': 'sf', 'orig_pkts': 1, 'orig_ip_bytes
': 42.0, 'resp_pkts': 0, 'resp_ip_bytes': 0}
[14:31:12] Ether / IP / ICMP 192.168.1.6 > 10.0.0.123 echo-reply 0
   Score: 1.000 → MALICIOUS
----------------------------------------
   Features: {'proto': 'other', 'service': 'unknown', 'duration': 0.0, 'orig_byt
es': 60.0, 'resp_bytes': 0.0, 'conn_state': 'sf', 'orig_pkts': 1, 'orig_ip_bytes
': 60.0, 'resp_pkts': 0, 'resp_ip_bytes': 0}
[14:31:12] Ether / IP / ICMP 10.0.0.123 > 192.168.1.6 echo-request 0 / Padding
   Score: 0.000 → benign
```

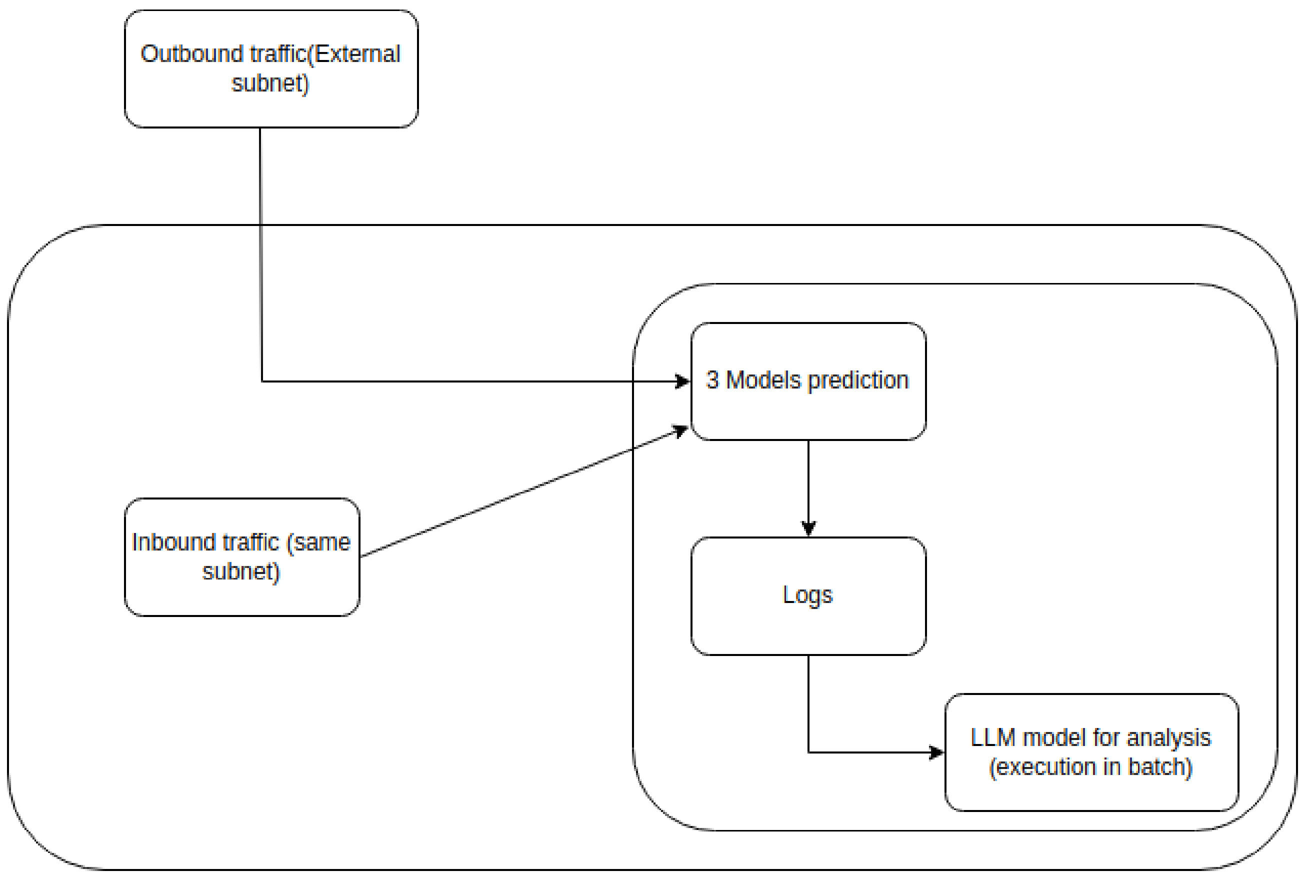
Attack simulation from the Kali VM



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -1 --flood -a 10.0.0.123 192.168.1.6

HPING 192.168.1.6 (eth0 192.168.1.6): icmp mode set, 28 headers + 0 data byte
s
hping in flood mode, no replies will be shown
```

Final architecture


Outbound traffic(External subnet)
3 Models prediction
Inbound traffic (same subnet)
Logs
LLM model for analysis (execution in batch)

For the execution steps , please check the demo videos existing in the readme file.