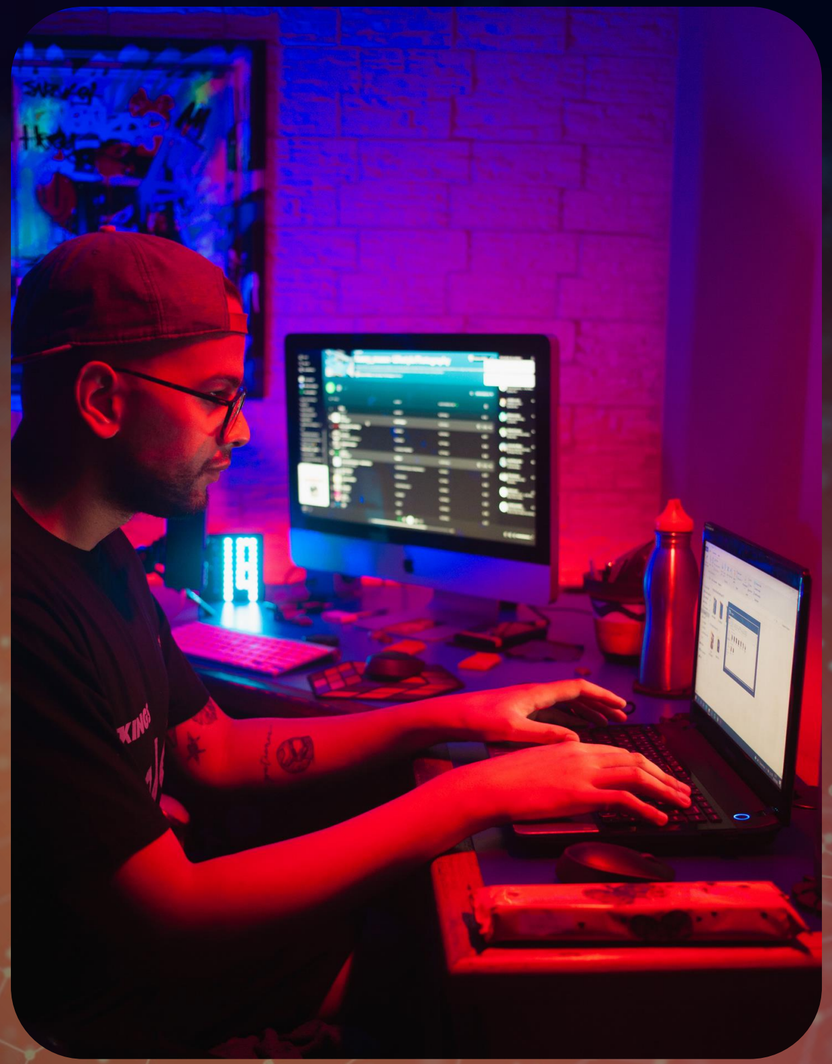
ML-Based Intrusion Detection for SDN-Based Broadband Networks

Presented by:

Cherifa Chaouch
Meryem Ben Salem
Skander Gritli
Mohamed Amine Hamdi
Sumaya Othmani



PROBLEM :

As SDN networks grow,

they become vulnerable to security threats.

→ IDS is required

to identify and mitigate attacks.

OBJECTIVE :

Build an IDS using ML models

→ Enhances security in SDN networks :

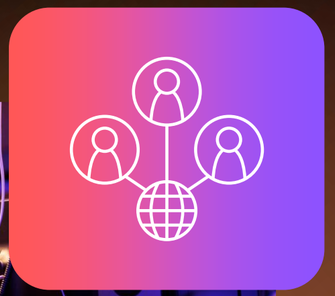
- Better data protection
- Uninterrupted service.

WHAT is SDN?

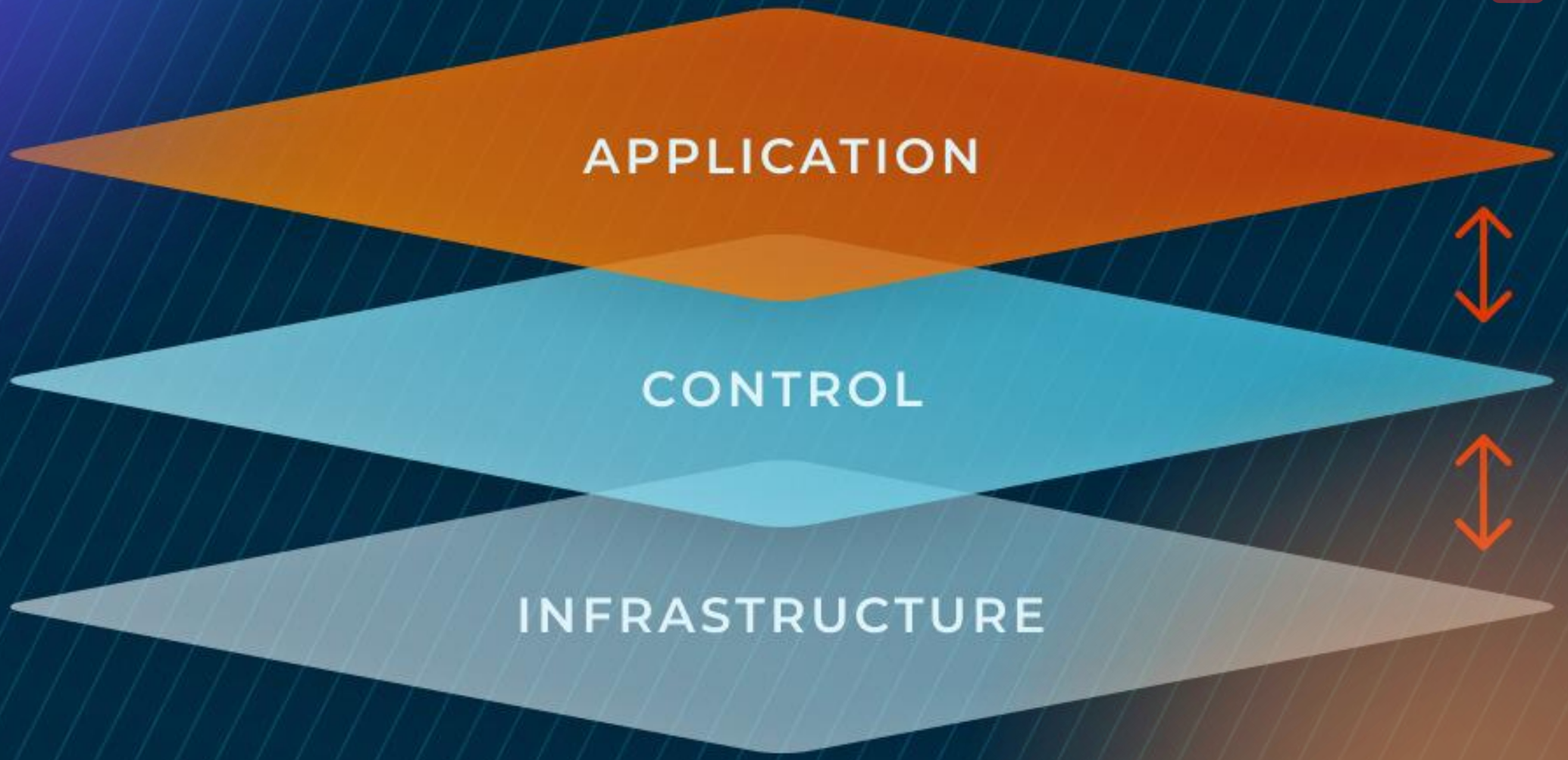
Separates the **control plane**

from the **data plane**

→→ A network more agile and programmable.



SDN

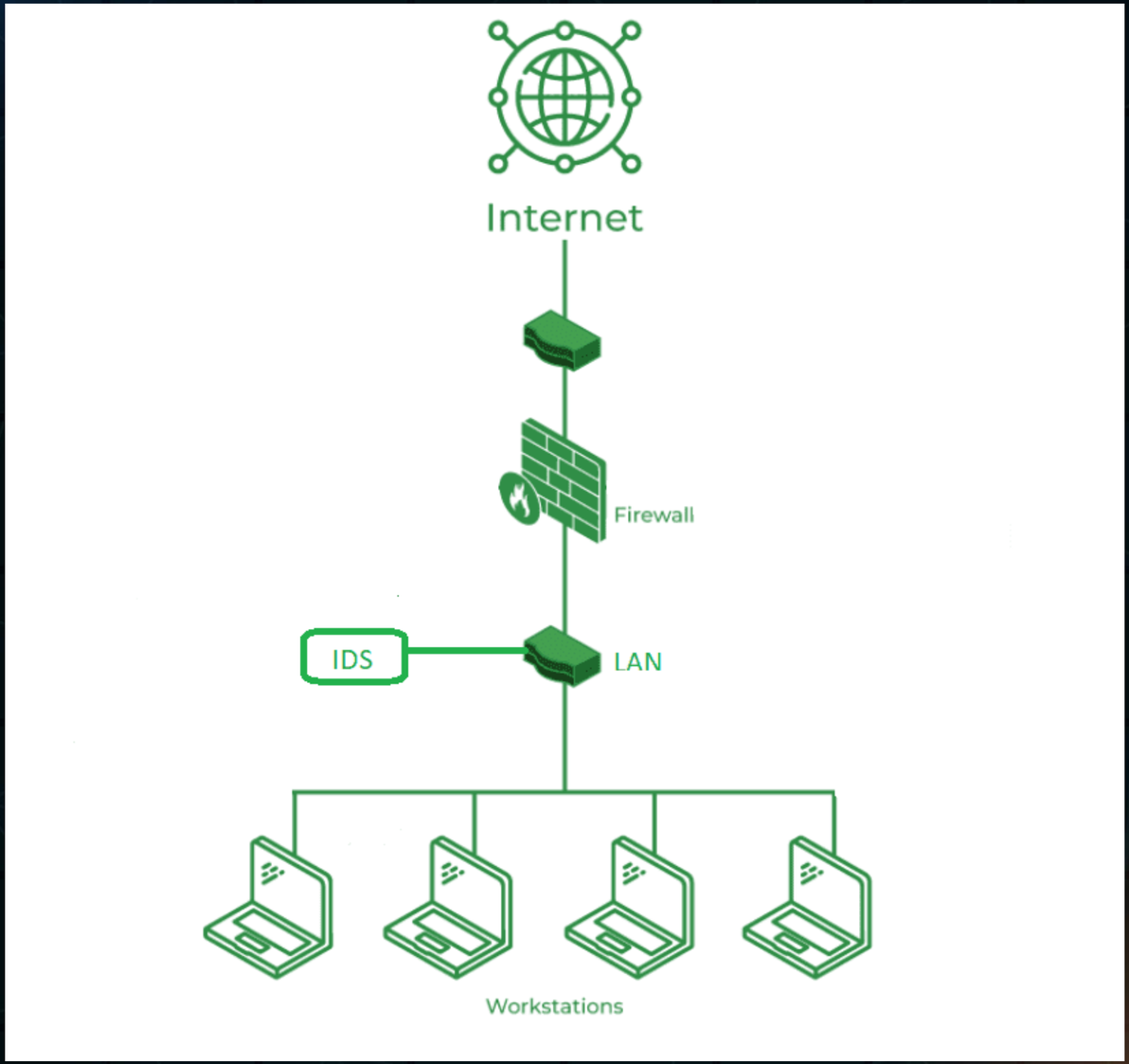


BENEFITS of IDS ?

Centralized Control

Dynamic Response

Visibility



Intrusion Detection System (IDS)

WHAT is the role of ML ?

SDNs are vulnerable to attacks

- **DDoS**
- **Spoofing**
- **...**

ML

↓

Threat Detection ++

STEP #1

ML MODEL

Dataset :

NSL-KDD

for evaluating IDS ,
containing network traffic data with labeled attacks
DoS, probe, U2R, R2L

NSL-KDD

261 New Notebook Download

Data Card Code (212) Discussion (2) Suggestions (0)

```
282,tcp,ftp,SF,160,599,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,37,0.15,0
0,tcp,http,SF,229,509,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,21,255,1.00,0.
0,tcp,imap4,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,280,8,0.00,0.00,1.00,1.00,0.03,0.06,0.00,255,8,0.03,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.
7428,tcp,telnet,SF,0,44,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,217,0.85
0,tcp,http,SF,282,396,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0
```

KDDTest1.jpg
KDDTrain+.arff
KDDTrain+.txt
KDDTrain+_20Percent.arff
KDDTrain+_20Percent.txt

Model #1 HFS-LGBM-based Approach

binary classification (Attack vs. No Attack)

Method for improving

feature selection & ***classification accuracy***

Model #1 HFS-LGBM-based Approach

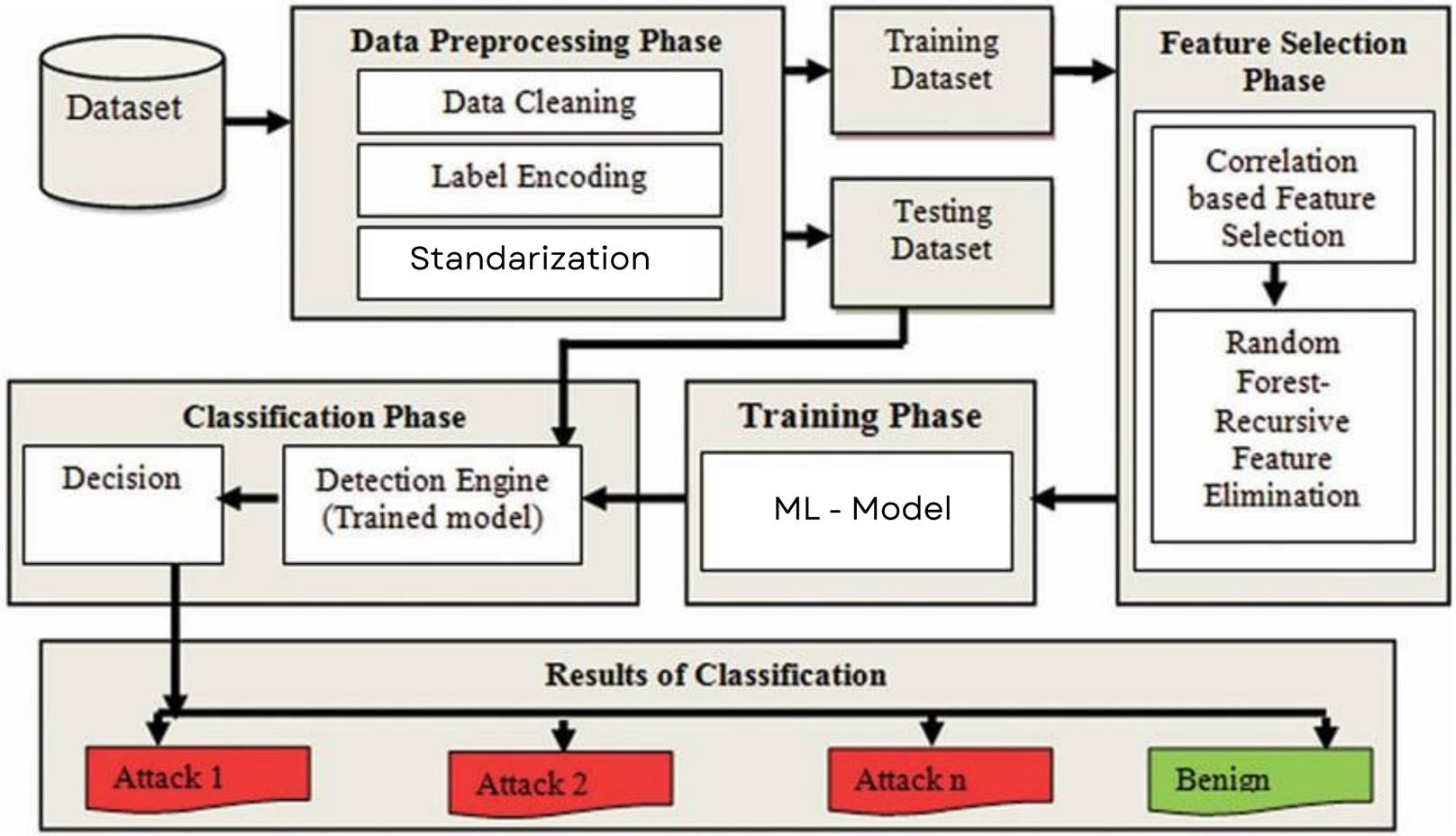


Figure 3: The proposed HFS-LGBM based IDS

Model #1 HFS-LGBM-based Approach

Data Preprocessing

Data cleaning

Missing values

Duplicates

Data Type Optimization

Feature scaling

Scale numerical features to standardize them

one-hot encoding to categorical features

HFS

Model #2 HFS-LGBM-based Approach

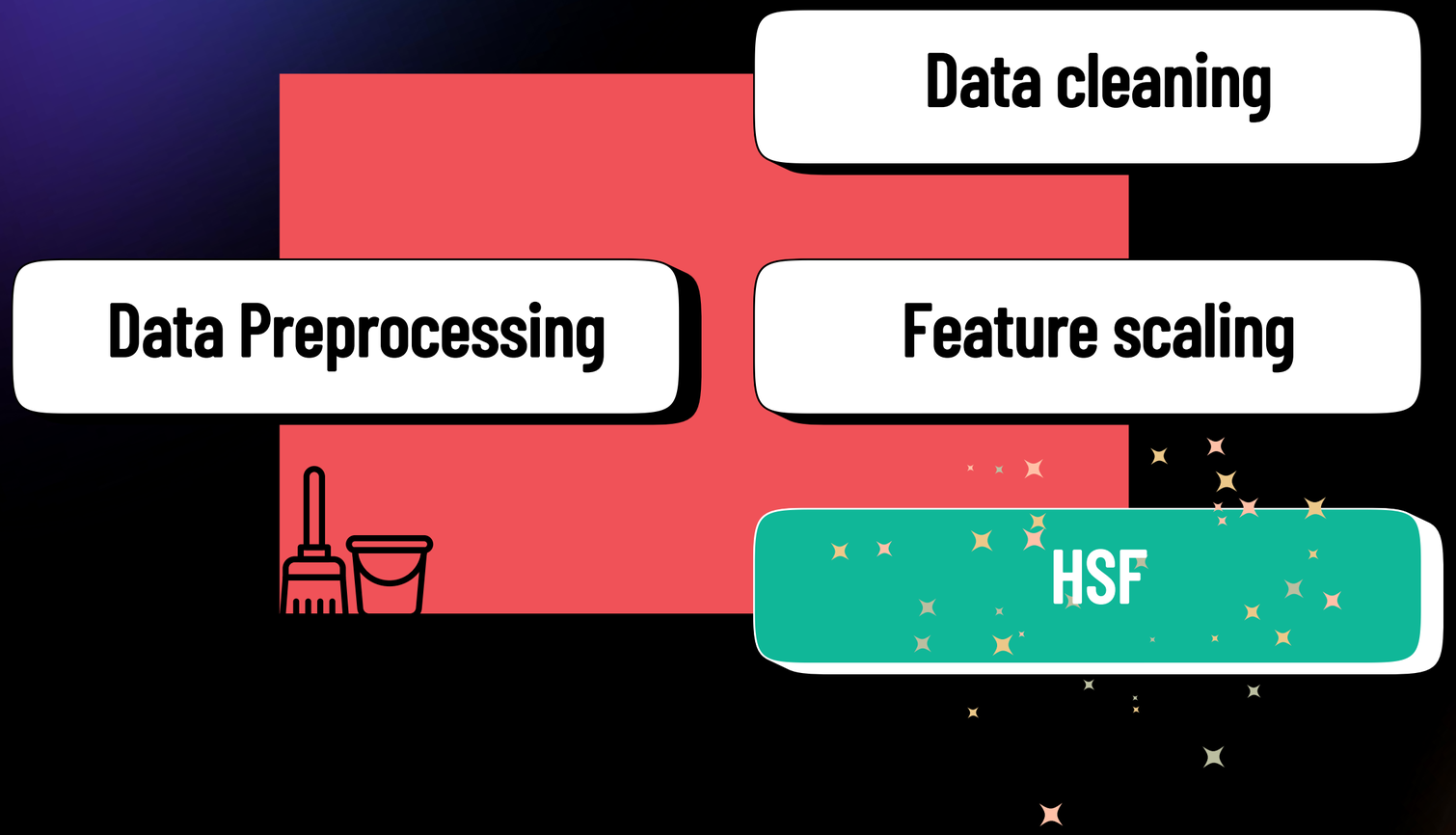
Data cleaning

Data Preprocessing

Feature scaling

HFS

Model #1 HFS-LGBM-based Approach



Hybrid Feature Selection =

CFS RF-RFE

Model #1 HFS-LGBM-based Approach

Random Forest Recursive Feature Elimination (RF-RFE)

+

Correlation-Based Feature Selection (CFS)

Model #1 HFS-LGBM-based Approach

RF-RFE

Eliminate the least important features

+

CFS

Remove redundant features from the dataset

Handling Class Imbalance:

Random Oversampling

replicates samples from the minority class

→ Ensure equal representation during model training

Model #1 HFS-LGBM-based Approach

Feature Engineering

CFS & RF-RFE

LightGBM

A **tree-based** boosting **algorithm** similar to XGBoost, (faster & more memory-efficient)

Model #1 HFS-LGBM-based Approach

Handling Class Imbalance

Random Oversampling

+

LGBM class weight adjustment for better balance

Model #1 HFS-LGBM-based Approach

Model Selection & Evaluation

With Key Metrics similar to the initial model, these where the results :

Model	Accuracy	Precision (0)	Recall (0)	F1-Score (0)	Precision (1)	Recall (1)	F1-Score (1)	General Recall	General F1-Score
SVM (linear)	0.7695	0.95	0.63	0.76	0.66	0.95	0.78	0.79	0.77
SVM (poly)	0.7943	0.96	0.66	0.79	0.68	0.97	0.80	0.82	0.79
SVM (rbf)	0.7909	0.97	0.65	0.78	0.68	0.97	0.80	0.81	0.79
SVM (sigmoid)	0.7202	0.86	0.60	0.71	0.63	0.88	0.73	0.74	0.72
Logistic Regression	0.7623	0.91	0.65	0.76	0.66	0.92	0.77	0.78	0.76
K-Nearest Neighbors	0.7822	0.97	0.64	0.77	0.67	0.98	0.79	0.81	0.78
Random Forest	0.7807	0.88	0.71	0.79	0.70	0.87	0.77	0.79	0.78
Decision Tree	0.8687	0.87	0.90	0.89	0.86	0.83	0.84	0.86	0.87
Gradient Boosting	0.7976	0.83	0.81	0.82	0.76	0.78	0.77	0.80	0.79
XGBoost	0.7972	0.84	0.80	0.82	0.75	0.79	0.77	0.80	0.79

Model #2 Network Intrusion Detection Pipeline

Classification by Type (Multiclass Classification)

Data Preprocessing

Data Preprocessing

Feature Engineering

Feature Extraction

Model #2 Network Intrusion Detection Pipeline

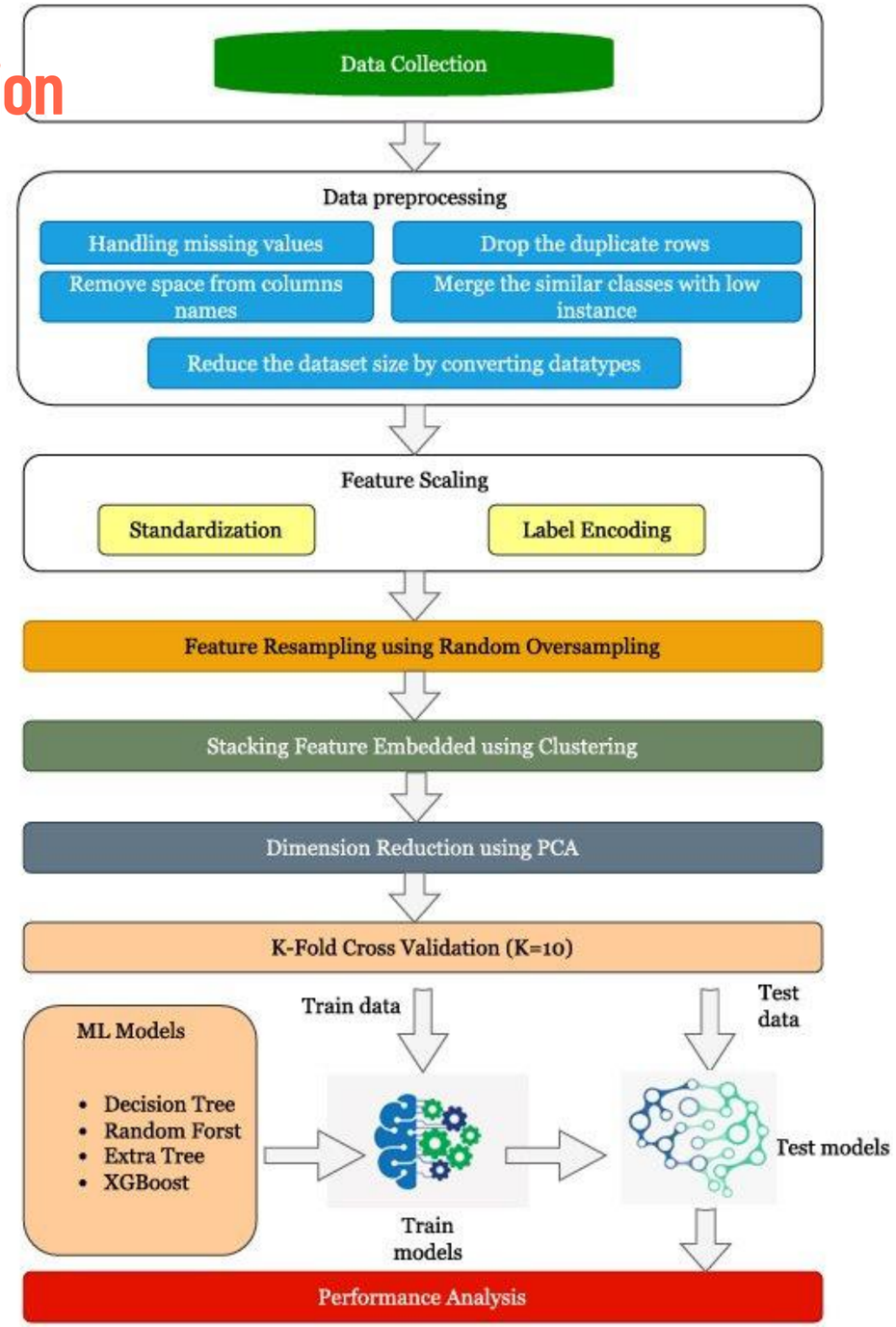


Fig. 1 Proposed framework for stacking feature embedded with PCA for intrusion detection

Model #2 Network Intrusion Detection Pipeline

Data Preprocessing

Data cleaning

Missing values

Duplicates

Data Type Optimization

Feature scaling

Scale numerical features to standardize them

one-hot encoding to categorical features

Handling Class Imbalance:

Random Oversampling

replicates samples from the minority class

→ Ensure equal representation during model training

Model #2 Network Intrusion Detection Pipeline

Feature Engineering

Clustering with K-Means

→ Create meta-features

→ Improve model performance

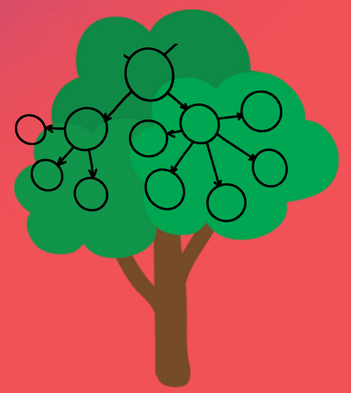
PCA for Dimensionality Reduction

Reduce the dimensionality of the dataset while preserving as much variance as possible

Model #2 Network Intrusion Detection Pipeline

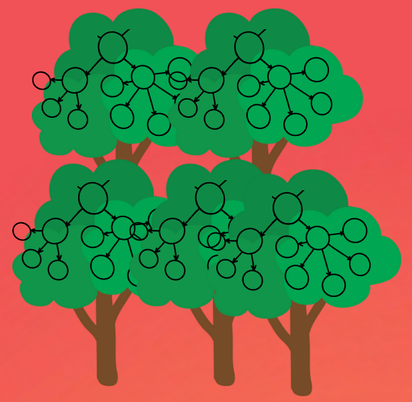
Model Selection & Evaluation

We selected three ML models to train and evaluate :



VS

VS



Decision Tree **Random Forest** **Extra Trees**

Model #2 Network Intrusion Detection Pipeline

Model Selection & Evaluation

Training with K-Fold Cross-Validation

Perform 10-fold cross-validation on the training dataset to:

- Split the dataset into 10 equal parts
- Train the model on 9 parts and validate it on the 10th part.
- Repeat this process 10 times with different splits

Model #2 Network Intrusion Detection Pipeline

Model Selection & Evaluation

Metrics Choice :

- Accuracy
- Precision
- Recall
- F1-score
- ROC Curve

```
Evaluating Decision Tree...
Decision Tree - Accuracy scores for each fold: [0.9996965568891271, 0.9995868008703007, 0.9997223818347333, 0.999664275
Decision Tree - Average accuracy: 0.9997
Evaluating Random Forest...
Random Forest - Accuracy scores for each fold: [0.9997675754895441, 0.9997869441987488, 0.9998256816171581, 0.999806312
Random Forest - Average accuracy: 0.9998
Evaluating Extra Trees...
Extra Trees - Accuracy scores for each fold: [0.9997934004351503, 0.9997740317259457, 0.9998192253807565, 0.99979985667
Extra Trees - Average accuracy: 0.9998
Evaluating XGBoost...
```

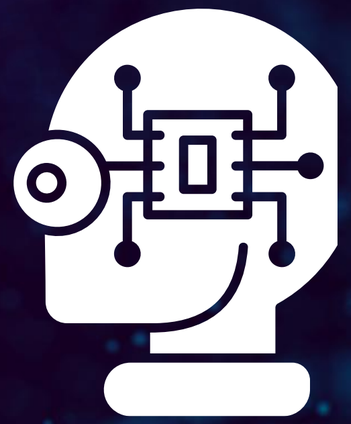
Model #2 Network Intrusion Detection Pipeline

Model Selection & Evaluation

and these where the results :

	Average Accuracy
Decision Tree	0.9997
Random Forest	0.9998
Extra Trees	0.9998

Model #1 vs Model #2



Model 1

HFS-LGBM-based Approach
(Binary Classification)

0,86

Model 2

Network Intrusion Detection Pipeline
(Multiclass Classification)

0,99

STEP #2

SDN INTEGRATION

Integrating SDN for **Real-Time Response**

After an attack is detected, SDN can trigger automated responses:

Rerouting Traffic

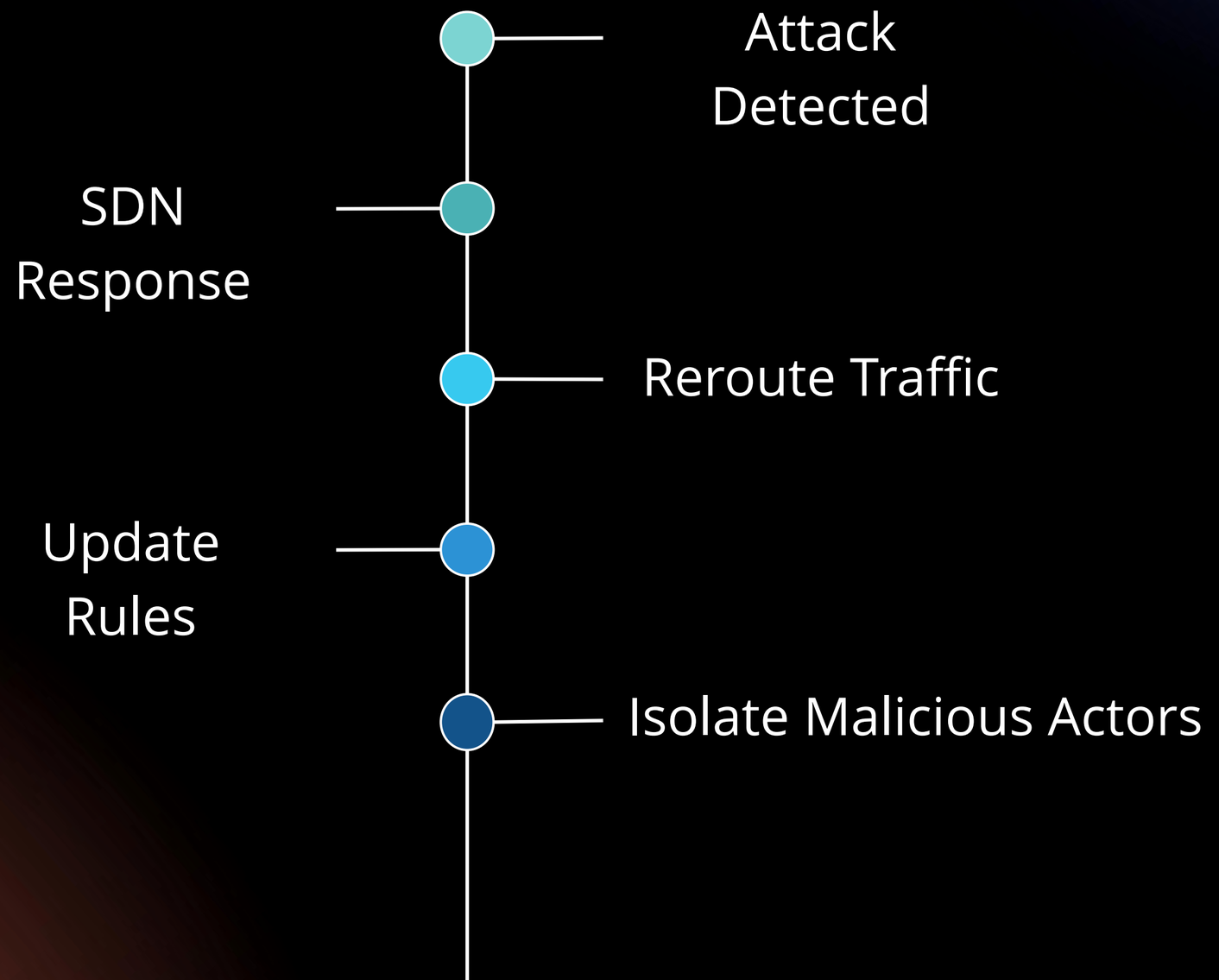
Isolate malicious traffic

Policy Adjustment

Change firewall rules or update network configuration

SDN enables continuous adaptation to new threats, keeping the network safe without manual intervention.

SDN Flowchart :



Real-World Impact of SDN Integration

Reduction in Attack Response Times

Cost Savings

Enhanced Network Efficiency

Scalability for Emerging Technologies

Improved Threat Intelligence

amine@ubuntu: ~ | amine@ubuntu: ~/karaf-0.21.1/bin

```
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=0.175 ms
64 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=0.171 ms
64 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=0.155 ms
64 bytes from 10.0.0.1: icmp_seq=15 ttl=64 time=0.138 ms
64 bytes from 10.0.0.1: icmp_seq=16 ttl=64 time=0.602 ms
64 bytes from 10.0.0.1: icmp_seq=17 ttl=64 time=0.188 ms
64 bytes from 10.0.0.1: icmp_seq=18 ttl=64 time=0.183 ms
64 bytes from 10.0.0.1: icmp_seq=19 ttl=64 time=0.223 ms
64 bytes from 10.0.0.1: icmp_seq=20 ttl=64 time=0.202 ms
64 bytes from 10.0.0.1: icmp_seq=21 ttl=64 time=0.198 ms
64 bytes from 10.0.0.1: icmp_seq=22 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=23 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=24 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=25 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=26 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=27 ttl=64 time=0
64 bytes from 10.0.0.1: icmp_seq=28 ttl=64 time=0
```

odl - VMware Workstation

File Edit View VM Tabs Help

odl

Activities XTerm 19 Dec 17:05

amine@ubuntu: ~

amine@ubuntu: ~ | amine@ubuntu: ~/karaf-0.21.1/bin | amine@ubuntu: ~ | amine@ubuntu: ~

```
64 bytes from 10.0.0.1: icmp_seq=66 ttl=64 time=0.466 ms
64 bytes from 10.0.0.1: icmp_seq=67 ttl=64 time=0.196 ms
64 bytes from 10.0.0.1: icmp_seq=68 ttl=64 time=0.186 ms
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
64 bytes fr
^C
--- 10.0.0.1 ping statistics ---
86 packets transmitted, 86 received, 0% packet loss, time 86835ms
rtt min/avg/max/mdev = 0.129/0.426/4.494/0.681 ms
mininet> xterm h2
mininet> xterm h1
mininet>
```

"Node: h1"

```
root@ubuntu:/home/amine# python3 detection.py
Prediction: this host is hacked
root@ubuntu:/home/amine#
```

"Node: h2"

```
root@ubuntu:/home/amine# sudo hping3 -S -p 80 --flood --rand-source 10.0.0.1
sudo: hping3: command not found
root@ubuntu:/home/amine# sudo hping3 -S -p 80 --flood --rand-source 10.0.0.1
HPING 10.0.0.1 (h2-eth0 10.0.0.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Rechercher Adresse العروب FRA 17:05

Combining ML and SDN for Optimal Security

ML for Detection, SDN for Prevention

$\longrightarrow$ Collaborative Approach

THANK YOU !