Reliability for Smart Healthcare: A Network Slicing Perspective

Group No : 13

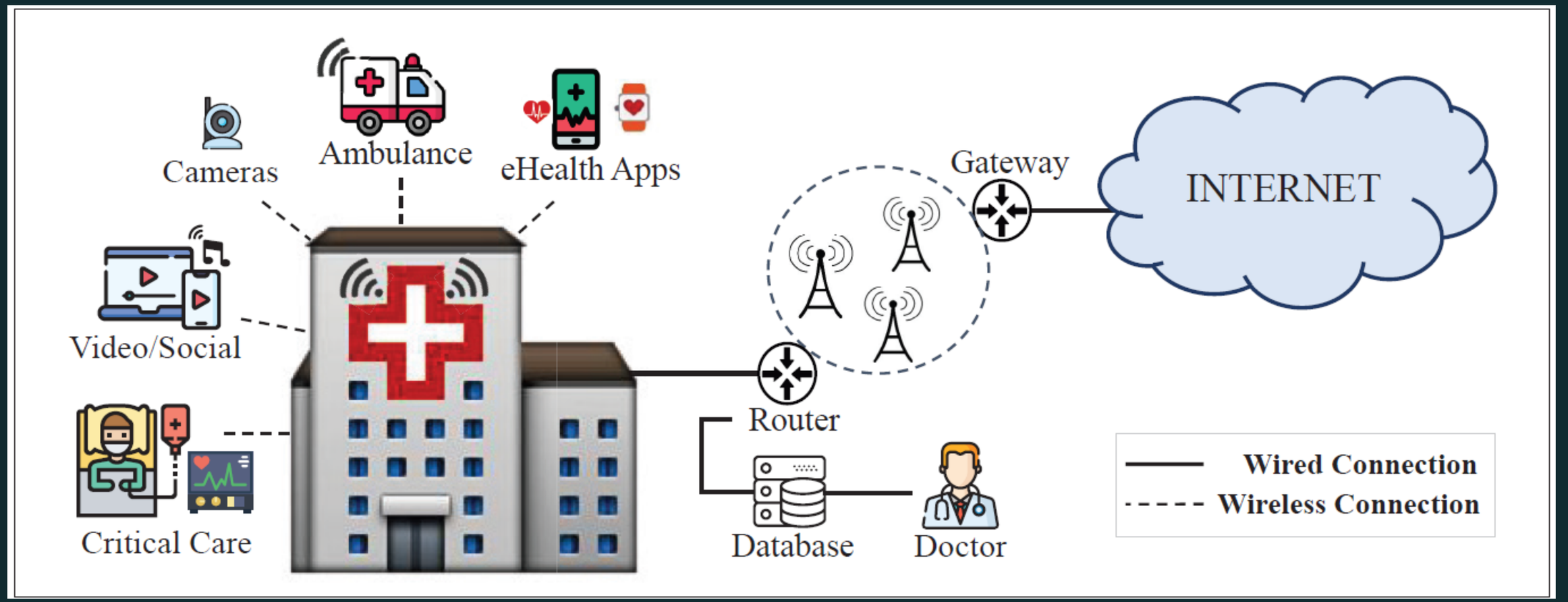
Group members:
Harshit mishra
K Kalyana Chakravarthi
Rakesh Roushan
Shashank Shekhar
Sunny kumar
Yash Dua

Problem statement

With the rapid advancement of information technology, smart health applications are emerging as one of the dependable pillars of the medical field, improving and enhancing treatment quality. However, all of these advantages result in a significant increase in network traffic, which causes significant delays and data loss, and frequently leads to network failure.
How can the network's quality and reliability be improved?

Aim

Network slicing aims at improving the quality of service for applications. Smart-health care requires efficient networking capabilities to provide low latency, low loss rate, and high reliability.
To achieve this we are using multi-class classification model to predict network slice for device or application.

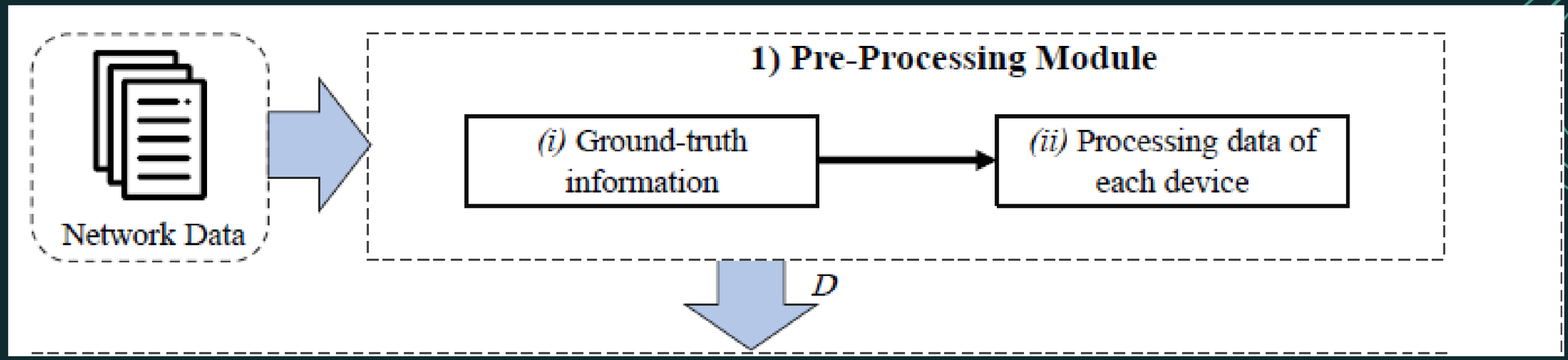


Smart hospital scenario

Model architecture :

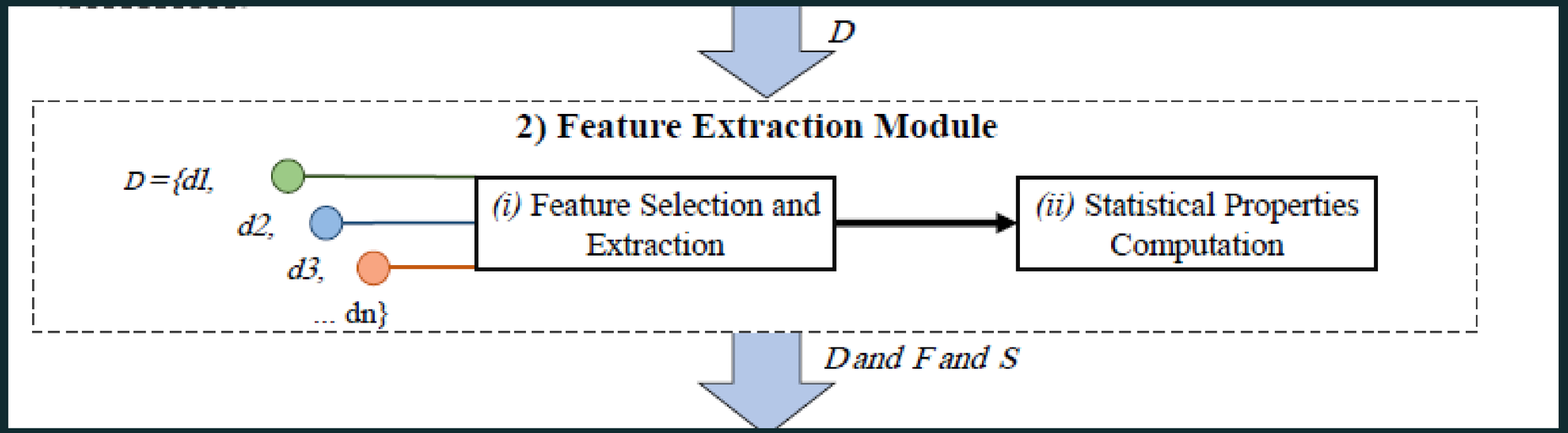
1. Pre-processing module
2. Feature Extraction module
3. Fingerprinting module
4. Network Slicing Configuration module

Pre- Processing Module



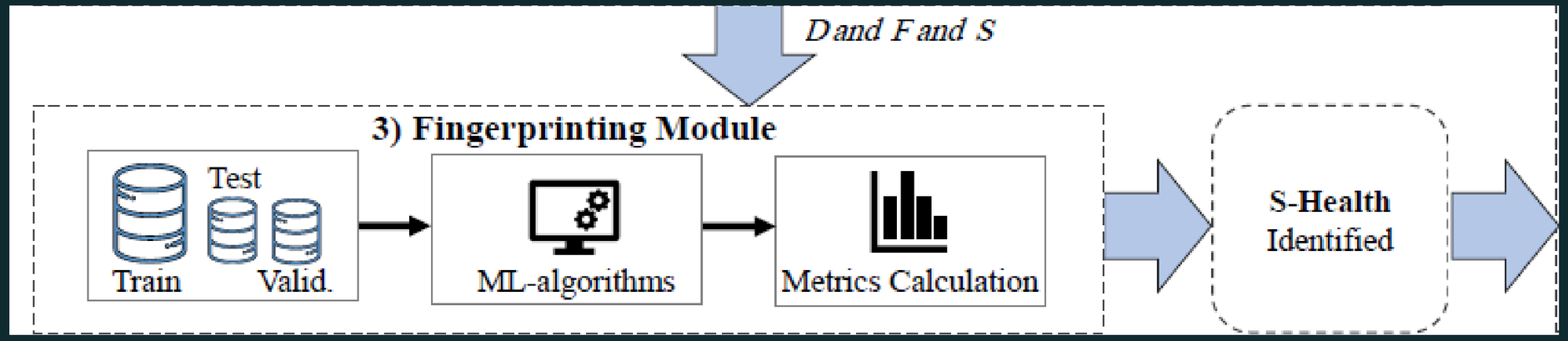
- It handles the network traffic in two components:
  the collection of ground truth information and data processing for each device.
- FLIPER uses the medium access control (MAC) address as ground truth to correctly indicate to which device each traffic flow belongs.
- Model creates a subset of network data for each class of devices, denoted by a set D = {d1, d2, ..., dn}, where each d means a specific device class with its network traffic.

Feature Selection Module



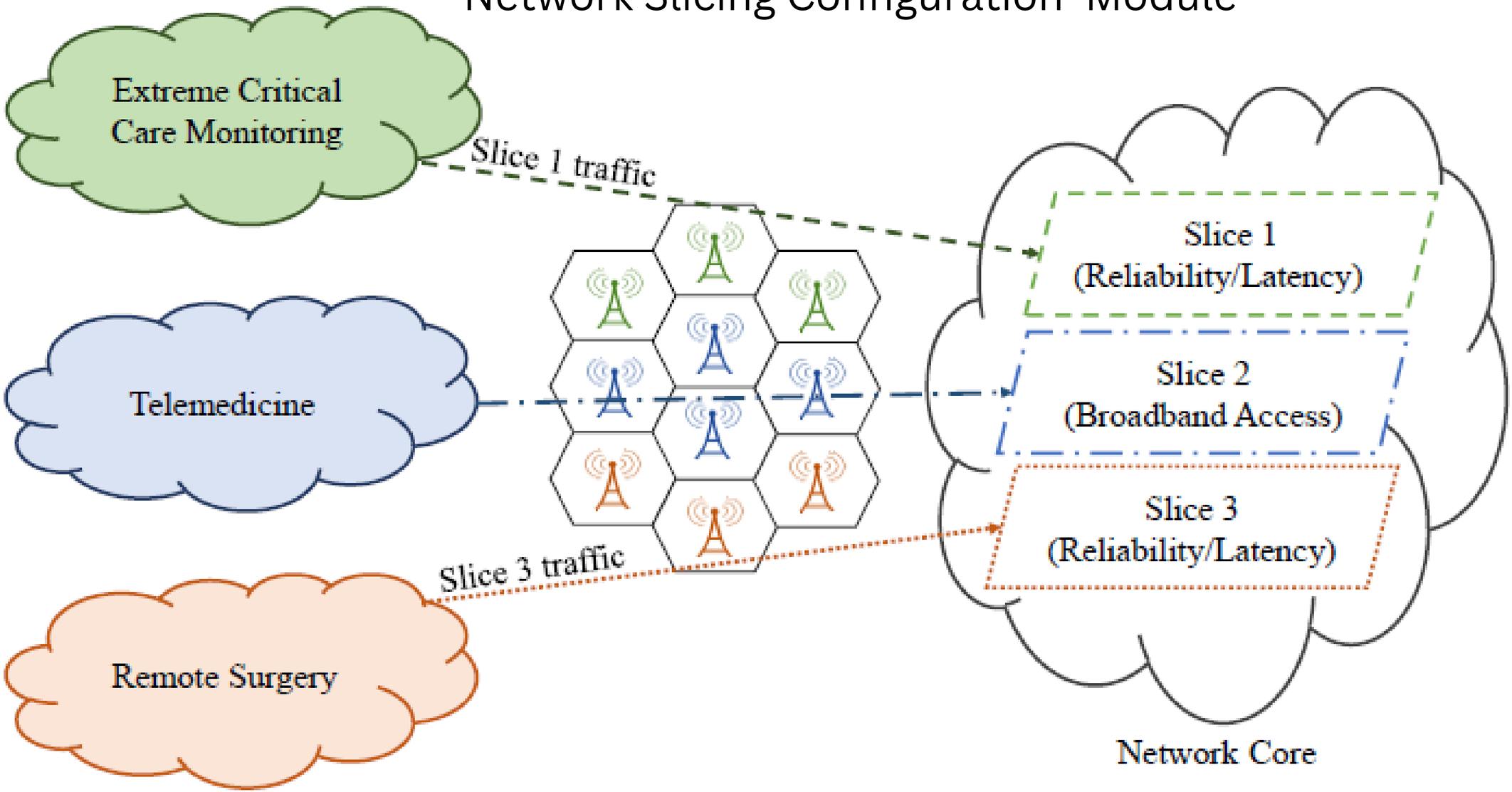
- FLIPER selects and extracts the features from the network traffic, defined as a set F of m network features, F = {f1, f2, ..., fm} that belongs to each class.
- For each set of features F, FLIPER calculates a set of x statistical measures, S = {s1, s2, ..., sx}. For instance, for the packet size feature, FLIPER computes the maximum (s1), minimum (s2), average (s3), and variance (s4).

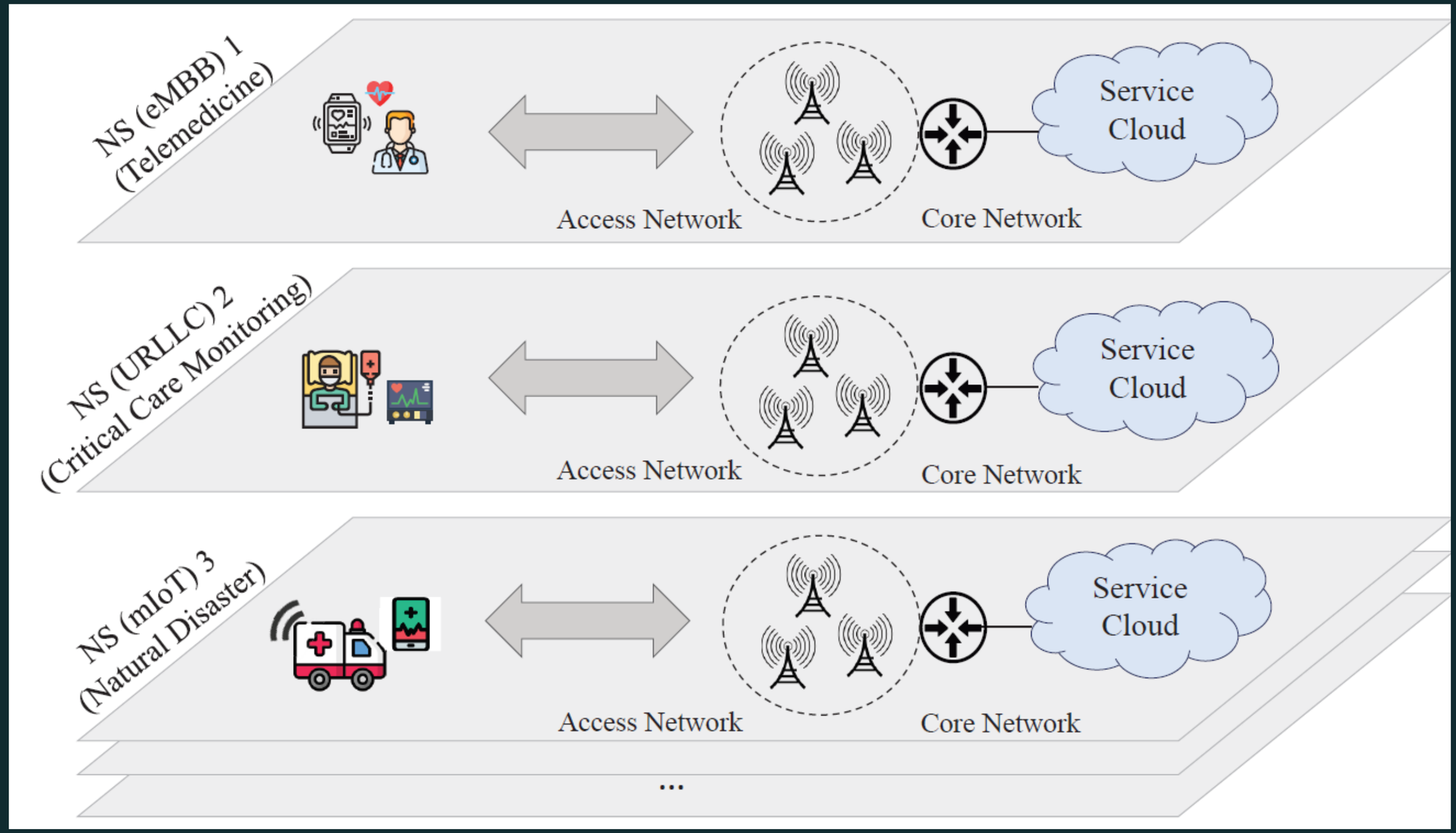
Fingerprinting Module



- This step uses distinct types of ML algorithms to fingerprint different s-health applications, for example, the Random Forest based on decision tree rules.
- It receives as input the network traffic features F and statistical properties S of each class of device. As each class is labeled, merges them in a final file, called training data.It supports unbalanced data .
- The fingerprinting of applications follows the holdout method, where the training data is divided into testing and validation data.After creating the final file, FLIPER applies the MLalgorithms and computes performance metrics (e.g., accuracy) to fingerprint the classes of applications.

Network Slicing Configuration Module





Example of network slices for s-health

eMBB(Enhanced Mobile Broadband) slice type

- eMBB traffic can be considered to be a direct extension of the 4G broadband service.
- It is characterized by large payloads and by a device activation pattern that remains stable over an extended time interval. This allows the network to schedule wireless resources to the eMBB devices such that no two eMBB devices access the same resource simultaneously.
- The objective of the eMBB service is to maximize the data rate, while guaranteeing a moderate reliability.

URLLC(Ultra-reliable Low-Latency Communication) slice type

Ultra-Reliable Low Latency Communications (URLLC), a subset of the 5G network architecture, ensures more efficient scheduling of data transfers, achieving shorter transmissions through a larger subcarrier, and even scheduling overlapping transmissions. It supports highly important data transfer that requires low latency, such as self-driving cars and remote surgery.
This will be used for mission-critical applications which require a guaranteed connection and low latency.

mIoT(Massive Internet of Things) slice type

Massive IoT is used when broad coverage is needed and in small data volumes instead of focusing on the connection’s speed. For example, these devices can be located in challenging radio conditions. So, connecting anything from hundreds to billions of devices and ensuring that data is transmitted is crucial. Given that 5G allows for this level of connectivity, there must be a way to ensure that the IoT devices have available connections at a reasonable cost to handle so much data running through them. There is no point in having super-fast networks if connectivity drops and the end-user is unhappy with the service.

Code

Thank You