CSI Path Trajectory Dataset: Enhancing Forensic Skills through Realistic Scenarios for enhancing tracking human using WiFi

The dataset collection process in the specified location with 13 paths involves using two Raspberry Pi (RPi) units functioning as receivers and a transmitter device in the form of a router. Additionally, the router is coupled with a laptop for data injection purposes. The collection process entails transmitting and receiving data signals across the 13 paths within the location. The RPi units, acting as receivers, capture and record the data signals transmitted by the router. Meanwhile, the laptop, connected to the router, injects data into the network, allowing for a comprehensive dataset to be collected. This dataset collection process enables the subsequent analysis and evaluation of various aspects of network performance, signal quality, and other relevant parameters in the specified location.

In this context, "D_1_1" represents "Path 1, Direction 1," indicating a specific path within the location dataset collection process. On the other hand, "D_1_2" refers to the same path but in the opposite direction.

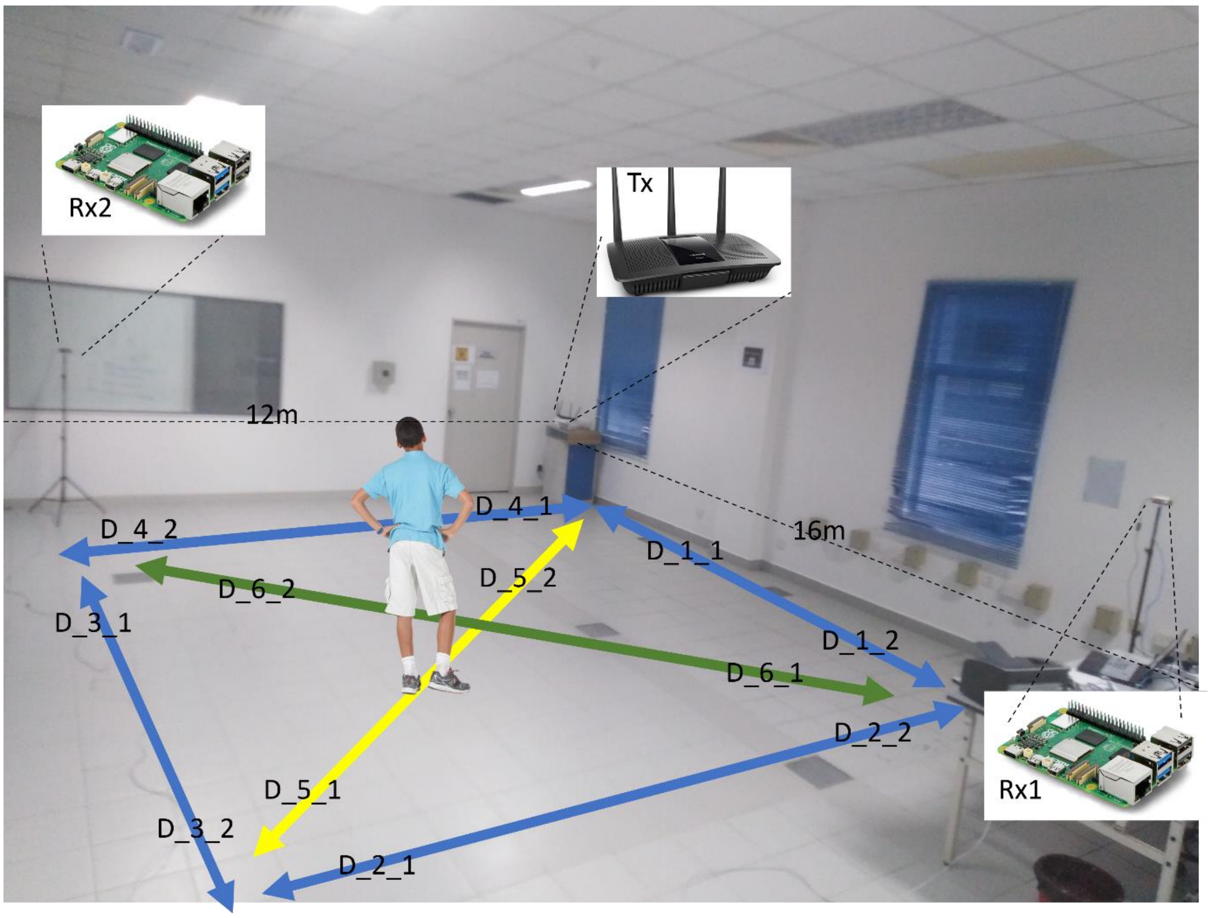


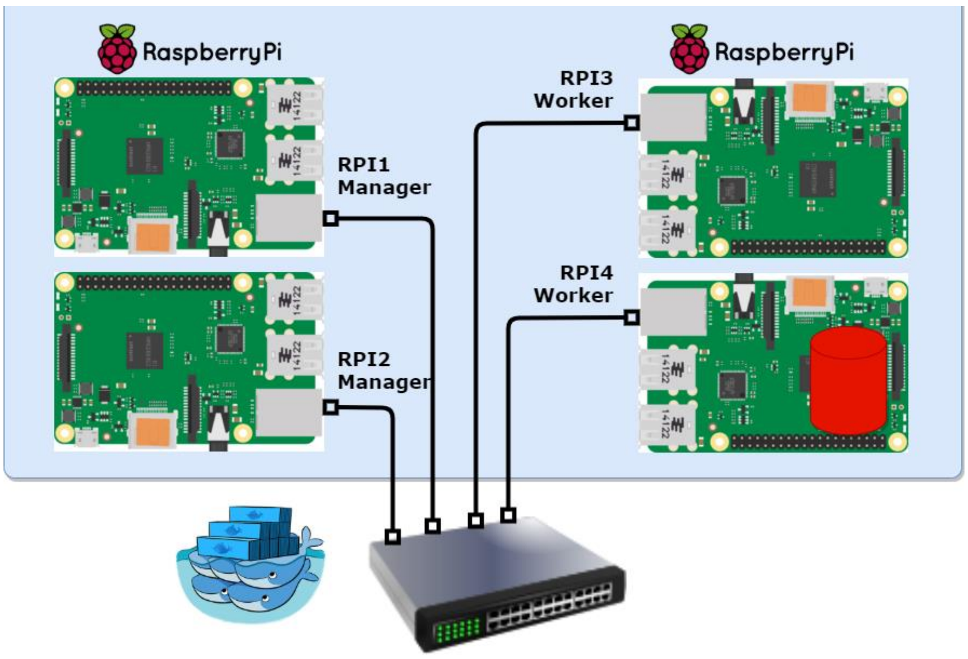
Figure 1 Layout of location Data collection

The dataset collection process involved the utilization of specific hardware and firmware components. Firstly, a Raspberry Pi 4B cluster, interconnected with a modem, was employed for data collection. This cluster of Raspberry Pi devices facilitated the simultaneous and coordinated capture of data from multiple points within the location.

Furthermore, a router was used as a target device to be monitored. The Nexmon firmware was employed to monitor the router, enabling the collection of relevant data and performance metrics from the router. Additionally, the router was injected with a Nitro 5 Laptop, which likely provided additional functionality or capabilities for data collection and analysis. The software tool used was MATLAB 2023a to collect and extract features from the data. MATLAB provided a comprehensive environment for processing and analyzing the captured dataset. Several supporting libraries and firmware were utilized to support the dataset collection process. The tcpdump was employed to collect the Channel State Information (CSI) dataset. The tcpdump is a widely used command-line tool for capturing and analyzing network traffic, including the CSI data.

The attached file encapsulates a noteworthy dataset consisting of 13 intricately curated paths, encompassing a variety of Crime Scene Investigation (CSI) scenarios. This collection goes beyond conventional trajectory datasets by incorporating instances of empty locations, thereby simulating realistic crime scene conditions. The dataset is accompanied by a Matlab file illustrating the application of a cutting-edge BiLSTM (Bidirectional Long Short-Term Memory) Recurrent Neural Network (RNN) for training purposes. This innovative approach in utilizing BiLSTM RNN within the Matlab environment enhances the dataset's potential for training forensic professionals, providing a robust platform for honing their investigative skills in diverse and challenging environments.

Clustered Raspberry Pi is used to enable multiple nodes to be connected running and monitored with one master device. In this experiment, we used only two Raspberry Pi which are connected with a switch that is connected directly to MATLAB to enable real-time data extracting and monitoring.

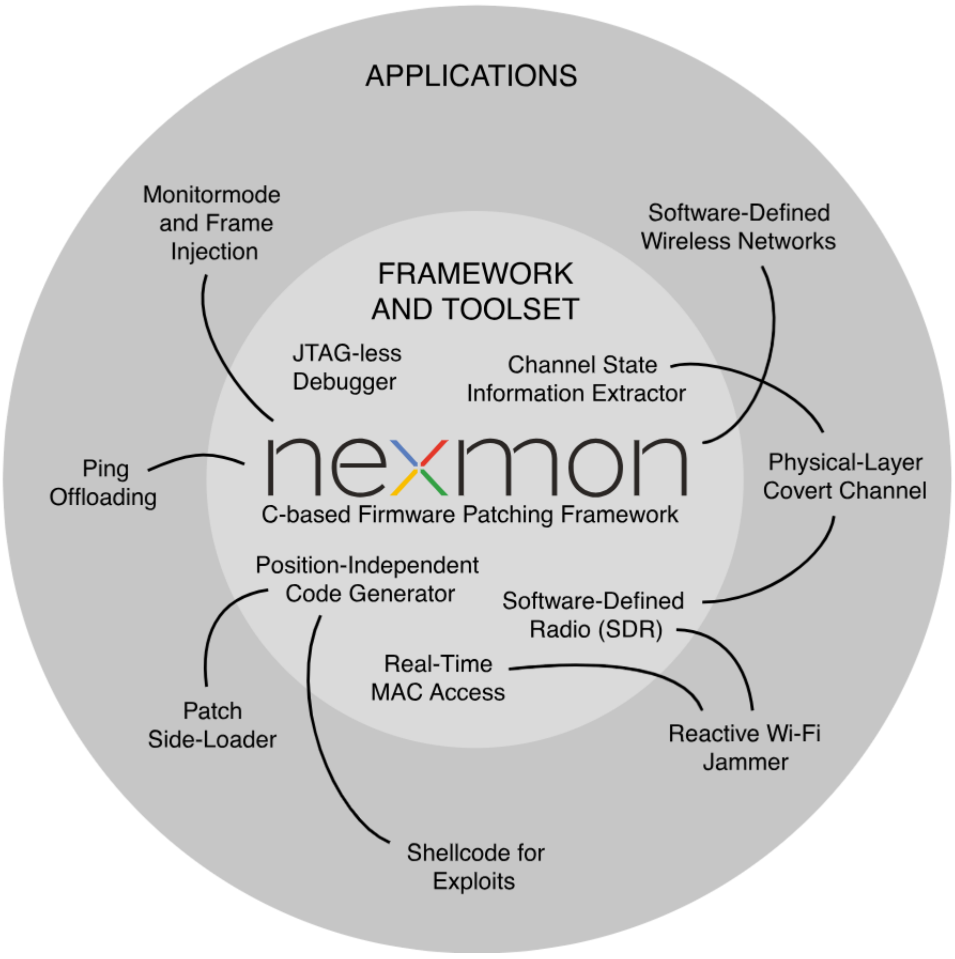


To set the Raspberry Pi to monitor mode, the following code steps were used:

```
sudo su
cd Desktop
cd fa
```

```
mcp-c 09/20-C 1-N 1-m 1C:3B:F3:2B:BD:53 -b 0x88
pkill wpa_supplicant
ifconfig wlan0 up
nexutil-Iwlan0-s500-b-l34-vCRABEQAAAQAcO/MrvVMAAAAAAAAAAAAAAAAAAAAAAAAAAA==
iw dev wlan0 interface add mon0 type monitor
ip link set mon0 up
tcpdump-i wlan0 dst port 5500-vv-w walking.pcap-c 200000
```

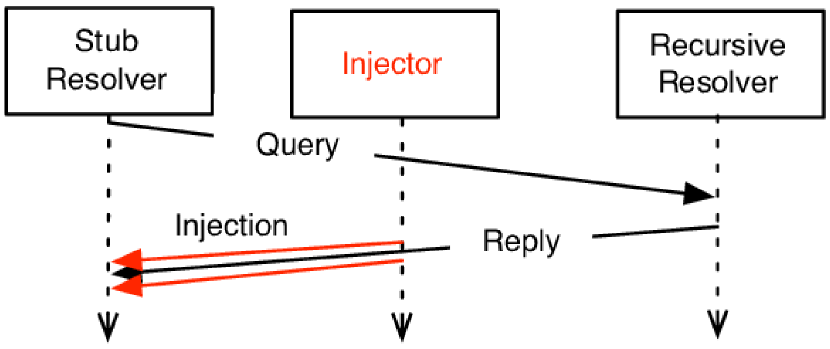
https://github.com/seemoo-lab/nexmon


APPLICATIONS
FRAMEWORK AND TOOLSET
Monitormode and Frame Injection
Software-Defined Wireless Networks
JTAG-less Debugger
Channel State Information Extractor
Ping Offloading
nexmon
C-based Firmware Patching Framework
Physical-Layer Covert Channel
Position-Independent Code Generator
Software-Defined Radio (SDR)
Real-Time MAC Access
Patch Side-Loader
Reactive Wi-Fi Jammer
Shellcode for Exploits

data injection is using python code to control the packet injection of the router that is been monitored with RPI devices using the following python code

```
import time, socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
txmax = 0
i = 0
while True:
   # data = getdatachunk()  # this is always stable and takes ~ 10ms with no exception
                    # for reproducibility, replaced with:
```

```
time.sleep(0.01)
data = b'ffffffffffffffffffffffffffffffff' * 1042
s = time.time()
sock.sendto(data, ("192.168.0.1", 5500))  # send UDP packet
txmax = max(txmax, time.time() - s)
if i % 100 == 0:
   print(txmax)
   txmax = 0
i += 1
```


Stub Resolver
Injector
Recursive Resolver
Query
Injection
Reply