Task 1

“Perform BGP analysis using PyBGPStream and detect anomalies in AS path lengths for IPv4 prefixes over two time periods: from January 15, 2023, from 12:00 to 13:00 and from 15:00 to 17:00"

Task 2

"Given a data.pcap file containing network traffic data, create a Python script using Scapy to process the pcap file and store this data in a pandas DataFrame and display it."

Task 3

Load data from '5G_Infrastructure.csv' and Generate Python code to calculate 5G network performance KPIs:Total Network Capacity, Capacity per Area…”

Output

```
bgp_data_15_17 =
pypbgpstream.load_bgp_data('
2020-01-15T15:00:00/2020-01-
15T17:00:00')
bgp_data_18_13 =
pypbgpstream.load_bgp_data('
2023-01-15T12:00:00/2023-01-
15T13:00:00')
…
```

Scoring

Incorrect Library Usage: The code attempts to use the 'PyBgpStream' library. However, it misuses the library in various aspects, including incorrect instantiation, property usage, and data extraction.

…

Score: 0

Output

```
import pybgpstream
import datetime

stream =
pybgpstream.BGPStream(
    from_time=start_time_1,
    until_time=end_time_1,
    filter="ipv4 prefix more
0",
)
…
```

Scoring

Syntax Errors:

The time format used s not compatible with the 'PyBgpStream' library.

The filter used for 'ipv4' prefixes is not in the correct syntax.

…

Score: 6

Output

```
interval_1_data = {}
interval_2_data = {}
for rec in interval_1:
    for elem in rec:
        prefix =
elem.fields['prefix']
        path_len =
len(elem.fields['as_path'])
…
```

Scoring

The code correctly uses the 'pybgpstream' library to fetch BGP data for the specified time intervals. It uses the library's features, such as defining time intervals, selecting projects, and accessing BGP data records correctly. There are no issues with library usage in this code.

…

Score: 10