I’ve fixed the dimensions of the MLCloud array. It’s now time=xxxx, y\_cross\_track=3, x\_along\_track=14 to keep things consistent with the other arrays (Radiance, Latitude, Longitude).

For the compression, I used zlib=True in my code but didn’t specify the level, so it defaulted to 4. Since NetCDF relies on the zlib library, the compression rates depend on the data. I ran tests on the l1c data (similar to l1r) for orbit 90 with different compression levels, and here’s how it performed:

* **Level 0**: 4.61s, 1500101.15KB
* **Level 1**: 12.94s, 403806.87KB
* **Level 2**: 12.81s, 395145.87KB
* **Level 3**: 13.36s, 388348.49KB
* **Level 4**: 14.95s, 345637.57KB
* **Level 5**: 16.09s, 336181.67KB
* **Level 6**: 19.22s, 331018.04KB
* **Level 7**: 20.54s, 330225.07KB
* **Level 8**: 33.16s, 328839.72KB
* **Level 9**: 55.75s, 328257.45KB

**Are They Similar?**

* **Same Compression Algorithm (zlib)**: Both C++ and Python use the zlib library in NetCDF.
* **Compression Level Ranges (0-9)**: They use the same zlib settings.

**Factors That Could Cause Differences:**

* **Shuffle Filter**: C++ gives you control (shuffle=1), while Python automatically applies the filter with zlib=True. The shuffle filter is especially useful for data types like floating-point values because it rearranges bytes to improve compression.

I created a script that reads the compressed NetCDF files and measures the time it takes to read them. The results:

Compression Level: 0 - Read Time: 1.42s

Compression Level: 1 - Read Time: 5.06s

Compression Level: 2 - Read Time: 4.81s

Compression Level: 3 - Read Time: 4.78s

Compression Level: 4 - Read Time: 4.92s

Compression Level: 5 - Read Time: 4.79s

Compression Level: 6 - Read Time: 4.72s

Compression Level: 7 - Read Time: 4.73s

Compression Level: 8 - Read Time: 4.70s

Compression Level: 9 - Read Time: 4.68s

* **Compression level affects** the read time because of the decompression process.
* **After reading** the file, the data is decompressed and behaves the same regardless of the compression level originally used.
* The primary impact of compression is on **file size** (disk storage) and **read/write times** (due to compression/decompression). Once the data is in memory, it's the same regardless of compression.

**Variables to Write**: We only write the variables Epoch, ISS\_Latitude, ISS\_Longitude, and Radiance

Compression Level: 0 - Time: 1.85s, File Size: 499964.55KB

Compression Level: 1 - Time: 7.03s, File Size: 281496.82KB

Compression Level: 2 - Time: 7.28s, File Size: 280522.96KB

Compression Level: 3 - Time: 7.56s, File Size: 279568.68KB

Compression Level: 4 - Time: 8.49s, File Size: 279878.77KB

Compression Level: 5 - Time: 9.08s, File Size: 279220.42KB

Compression Level: 6 - Time: 9.47s, File Size: 278545.00KB

Compression Level: 7 - Time: 9.78s, File Size: 278401.27KB

Compression Level: 8 - Time: 13.18s, File Size: 278100.36KB

Compression Level: 9 - Time: 19.87s, File Size: 278011.44KB

1422 images, 155 MB

A black and white image of a grey circle with white text

Description automatically generated

# Confusion Matrix

train\_size = int(data\_size \* 0.7)

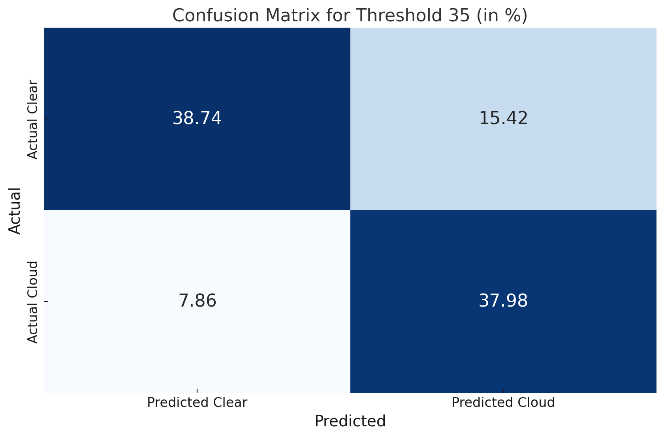
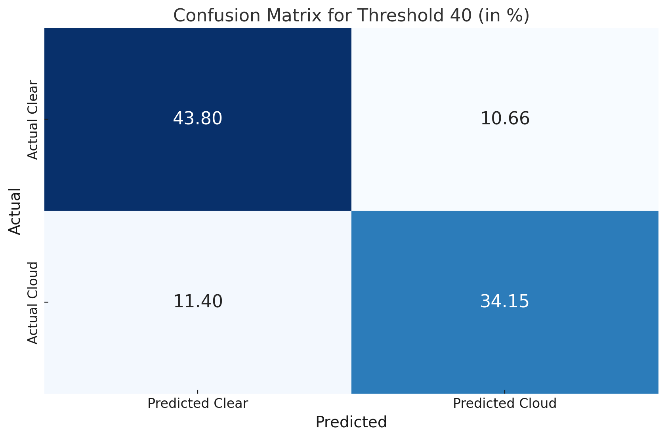
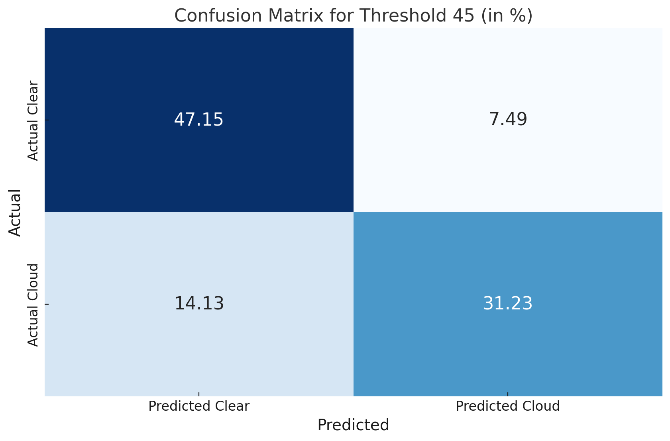
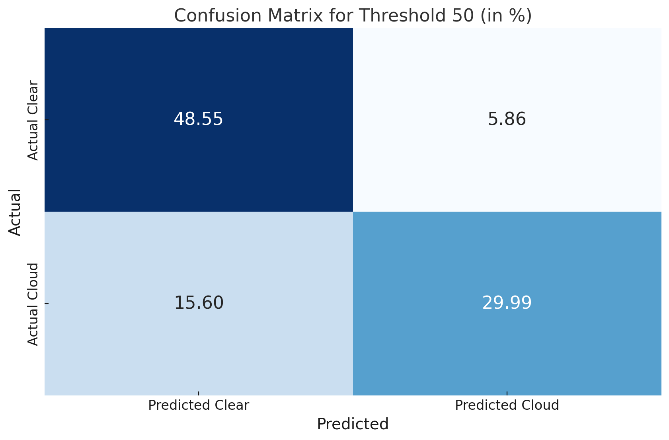
val\_size = int(data\_size \* 0.2)

test\_size = int(data\_size \* 0.1)

Validation Accuracy: ~80%

Apply the model to the test data.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threshold | TN | FP | FN | TP | TN | FP | FN | TP | Acc |
| 30 | 1504 | 875 | 274 | 1699 | 34.6% | 20.1% | 6.3% | 39.0% | 73.6% |
| 31 | 1523 | 830 | 288 | 1711 | 35.0% | 19.1% | 6.6% | 39.3% | 74.3% |
| 32 | 1571 | 787 | 293 | 1701 | 36.1% | 18.1% | 6.7% | 39.1% | 75.2% |
| 33 | 1629 | 733 | 329 | 1661 | 37.4% | 16.8% | 7.6% | 38.2% | 75.6% |
| 34 | 1670 | 699 | 339 | 1644 | 38.4% | 16.1% | 7.8% | 37.8% | 76.2% |
| **35** | **1686** | **671** | **342** | **1653** | **38.7%** | **15.4%** | **7.9%** | **38.0%** | **76.7%** |
| 36 | 1747 | 629 | 385 | 1591 | 40.1% | 14.5% | 8.8% | 36.6% | 76.7% |
| 37 | 1807 | 567 | 415 | 1563 | 41.5% | 13.0% | 9.5% | 35.9% | 77.4% |
| 38 | 1823 | 541 | 449 | 1539 | 41.9% | 12.4% | 10.3% | 35.4% | 77.3% |
| 39 | 1856 | 516 | 458 | 1522 | 42.6% | 11.9% | 10.5% | 35.0% | 77.6% |
| **40** | **1906** | **464** | **496** | **1486** | **43.8%** | **10.7%** | **11.4%** | **34.1%** | **77.9%** |
| 41 | 1926 | 438 | 528 | 1460 | 44.3% | 10.1% | 12.1% | 33.5% | 77.8% |
| 42 | 1940 | 411 | 548 | 1453 | 44.6% | 9.4% | 12.6% | 33.4% | 78.0% |
| 43 | 1984 | 388 | 562 | 1418 | 45.6% | 8.9% | 12.9% | 32.6% | 78.2% |
| 44 | 2013 | 349 | 573 | 1417 | 46.3% | 8.0% | 13.2% | 32.6% | 78.9% |
| 45 | 2052 | 326 | 615 | 1359 | 47.2% | 7.5% | 14.1% | 31.2% | 78.4% |
| 46 | 2050 | 320 | 612 | 1370 | 47.1% | 7.4% | 14.1% | 31.5% | 78.6% |
| 47 | 2075 | 295 | 634 | 1348 | 47.7% | 6.8% | 14.6% | 31.0% | 78.7% |
| 48 | 2082 | 279 | 649 | 1342 | 47.8% | 6.4% | 14.9% | 30.8% | 78.6% |
| 49 | 2121 | 246 | 657 | 1328 | 48.7% | 5.7% | 15.1% | 30.5% | 79.2% |
| 50 | 2113 | 255 | 679 | 1305 | 48.6% | 5.9% | 15.6% | 30.0% | 78.6% |
| 51 | 2129 | 243 | 710 | 1270 | 48.9% | 5.6% | 16.3% | 29.2% | 78.1% |
| 52 | 2127 | 229 | 718 | 1278 | 48.9% | 5.3% | 16.5% | 29.4% | 78.3% |
| 53 | 2147 | 226 | 717 | 1262 | 49.3% | 5.2% | 16.5% | 29.0% | 78.3% |
| 54 | 2157 | 211 | 739 | 1245 | 49.6% | 4.8% | 17.0% | 28.6% | 78.2% |
| 55 | 2174 | 198 | 749 | 1231 | 50.0% | 4.5% | 17.2% | 28.3% | 78.3% |
| 56 | 2173 | 186 | 762 | 1231 | 49.9% | 4.3% | 17.5% | 28.3% | 78.2% |
| 57 | 2196 | 172 | 795 | 1189 | 50.5% | 4.0% | 18.3% | 27.3% | 77.8% |
| 58 | 2196 | 172 | 813 | 1171 | 50.5% | 4.0% | 18.7% | 26.9% | 77.4% |
| 59 | 2210 | 161 | 808 | 1173 | 50.8% | 3.7% | 18.6% | 27.0% | 77.8% |
| 60 | 2215 | 153 | 843 | 1141 | 50.9% | 3.5% | 19.4% | 26.2% | 77.1% |
| 61 | 2218 | 145 | 848 | 1141 | 51.0% | 3.3% | 19.5% | 26.2% | 77.2% |
| 62 | 2240 | 134 | 848 | 1130 | 51.5% | 3.1% | 19.5% | 26.0% | 77.5% |
| 63 | 2254 | 114 | 874 | 1110 | 51.8% | 2.6% | 20.1% | 25.5% | 77.3% |
| 64 | 2251 | 116 | 880 | 1105 | 51.7% | 2.7% | 20.2% | 25.4% | 77.1% |
| 65 | 2256 | 116 | 894 | 1086 | 51.8% | 2.7% | 20.5% | 25.0% | 76.8% |
| 66 | 2265 | 111 | 920 | 1056 | 52.0% | 2.6% | 21.1% | 24.3% | 76.3% |
| 67 | 2262 | 109 | 939 | 1042 | 52.0% | 2.5% | 21.6% | 23.9% | 75.9% |
| 68 | 2268 | 108 | 946 | 1030 | 52.1% | 2.5% | 21.7% | 23.7% | 75.8% |
| 69 | 2274 | 84 | 973 | 1021 | 52.3% | 1.9% | 22.4% | 23.5% | 75.8% |
| 70 | 2281 | 91 | 994 | 986 | 52.4% | 2.1% | 22.8% | 22.7% | 75.1% |



Candidacy Exam Written Report and Oral Presentation.