Satellite Observation Success Prediction (SatNOGS Domain Study)

This document reviews the SatNOGS satellite ground-station project and relevant factors affecting observation success. SatNOGS (Satellite Networked Open Ground Station) is a global, open-source network of modular ground stations for tracking satellites[1]. Operators worldwide can use its web interface to schedule observations on any station and share results openly[2][3]. The SatNOGS DB provides a crowdsourced transmitter database (with free CC-BY-SA licensing) and a REST API for satellite/transmitter metadata[4][5]. Similarly, the SatNOGS Network API allows retrieving scheduled passes and observation results (also CC-BY-SA)[6]. Together, these resources let us fetch real pass parameters (e.g. elevation, duration, time) and observation outcomes for machine learning.

- **Project Goals:** Build a Python pipeline that uses SatNOGS pass metadata to predict whether a satellite observation will be successful. Apply classification models (e.g. logistic regression, random forests) to labeled SatNOGS data, and visualize results in an interactive Streamlit dashboard. Maintain code with GitHub/GitHub Actions and present findings clearly for SatNOGS contributors and academics.

Figure: A SatNOGS ground-station antenna (Aalto-1, Finland) used to receive satellite signals. SatNOGS relies on many such stations worldwide to track LEO satellites[1][2]. The SatNOGS network is fully open-source and participatory. Its design is modular – stations use readily available hardware (VHF/UHF antennas, SDRs, Raspberry Pi controllers, etc.)[1][7]. Via the SatNOGS web interface, any observer can schedule a pass on any station, leveraging the full network for broader coverage[2]. All resulting observations (waterfalls, audio, decoded packets) are made public under Creative Commons licensing[3], enabling this machine learning study.

Satellite Pass Characteristics

A **satellite pass** is the interval when a satellite rises above the local horizon and is within line-of-sight of a ground station[8]. Pass parameters include start/end times, duration, maximum elevation angle, and azimuth. In Low-Earth Orbit (LEO) scenarios, passes are short (often minutes) and rapidly changing. Key pass features influencing reception include:

- **Elevation angle:** the vertical angle above the horizon. Higher elevation means the satellite is more overhead[9]. High-elevation passes yield stronger signals and more favorable geometry because the path is more direct and has fewer obstructions[10][11]. By contrast, at low elevations (satellite near horizon) path loss is much greater[11].
- **Duration:** how long the satellite remains above the horizon. Longer passes give more time to capture data and integrate weak signals. Pass duration depends on orbital inclination and station latitude; it peaks when the station lies directly under the satellite's ground track[12].
- **Time-of-day:** (Local time or sun angle) can affect noise and visibility. For example, passes during daylight may have more solar radio interference on some bands, while night passes might see different ionospheric noise. We include time-of-day as a feature, although its impact is generally context-dependent.

In summary, passes that are longer and reach higher elevation generally produce better link quality, because path loss is minimized and antennas have clearer views[9][11].

Observation Success Criteria

SatNOGS observations are automatically labeled based on whether the satellite's signal was detected. A **"Good"** observation means the system identified the expected satellite signal (waterfall pattern, beacon, or data) in the recorded data[13]. A **"Bad"** observation means no signal was found in the data despite a functioning station[13][14]. (SatNOGS also uses "Failed" for cases where the station had a clear error or did not return data[15].)

In practice, SatNOGS contributors equate "Good" with *observed signal from target* and "Bad" with *no signal from target while station was operational*[14]. These labels come from vetting algorithms and user feedback, but for our purposes they serve as the ground-truth classes. Our ML pipeline will treat **Success = Good** (satellite heard) and **Failure = Bad** (no signal) when training classifiers.

Data Sources and Feature Engineering

We will gather data via the SatNOGS APIs and database:

- **SatNOGS Network API:** provides scheduled observation records including pass metadata (start/end UTC, duration, max elevation, station ID, satellite/transmitter ID, etc.) and the final automated/vetted status[6][13].
- **SatNOGS DB API:** provides reference details about satellites and transmitters (frequency, mode, which can be cross-referenced with passes)[5].

All these APIs are open and CC-BY-SA licensed, allowing free data collection[6][5]. Using Python (with `requests` or `pysatnogs` packages), we will download historical observations. Each record will include features like **max elevation**, **pass duration**, **start time of day**, plus details of the transmitter (frequency, encoding mode) and ground station (antenna gain patterns or site location if available). Categorical features (e.g. mode, station) can be one-hot encoded or similarly processed. Numerical features may be scaled or binned as needed.

Data will be cleaned to remove obvious anomalies (e.g. zero-duration passes or "Failed" cases). We will then split the dataset into training and test sets (e.g. 80/20), ensuring the class balance is maintained. Since "Good/Bad" rates may be imbalanced (many passes yield no signal), we may apply techniques like class weighting or SMOTE.

Classification Pipeline

We will experiment with several supervised classifiers using scikit-learn and related libraries. The pipeline steps include:

1. **Training/Test Split:** Partition the processed dataset into train and test subsets, preserving class proportions.
2. **Model Selection:** Evaluate multiple algorithms (e.g. logistic regression, random forest, gradient-boosted trees) to find the best predictor.
3. **Training:** Fit models on the training data. Use cross-validation and hyperparameter tuning (e.g. grid search) to optimize performance.
4. **Evaluation:** Measure accuracy, precision, recall, F1-score, and ROC-AUC on the test set. A confusion matrix will show how well "Good" vs "Bad" passes are

separated. Feature importance (from trees or coefficients) will reveal which pass parameters most influence success.

5. **Iterate and Refine:** Based on results, we may engineer additional features (e.g. antenna elevation restrictions, horizon mask, seasonal effects) or try ensemble methods to improve accuracy.

Key predictors likely include **elevation** and **duration** (higher values increase success chance) and possibly **time-of-day** or **satellite mode** if relevant. For example, signals on certain modes (e.g. CW beacons vs FM voice) might be easier/harder to detect, which our model can learn.

Interactive Dashboard and Tools

To make the results accessible to ground-station operators, we will build a Streamlit-based dashboard. This app will allow users to input or select upcoming passes (with parameters) and view the predicted success probability. It may also display visual summaries (charts of feature vs success rates, histograms, or model ROC curves) to help operators understand the predictions. Streamlit is chosen for its ease of integration with Python ML code and quick GUI development.

Languages & Libraries: Implementation will be in Python. Core libraries include Pandas and NumPy for data handling, scikit-learn for modeling, and Matplotlib/Seaborn for plots. The dashboard uses Streamlit. We will version-control the code on GitHub, using GitHub Actions for continuous integration (e.g. running tests or linting on push). Documentation (this README and any Jupyter notebooks) will be maintained in the repo.

Reproducibility: All data processing and model steps will be documented. The SatNOGS data API access requires an API key (obtainable via a SatNOGS account), and our code will include instructions to fetch data. Any computed results (trained model, plots) will be generated automatically by scripts or notebooks.

Expected Outcomes

By correlating historical pass metadata with observation outcomes, we expect to identify patterns and build a reliable predictor of success. The final model can help planners decide which passes to schedule (e.g. prioritizing high-probability opportunities) and may guide station operators on improvements (e.g. antenna upgrades for low-elevation coverage). All results will be presented in a clear, professional manner with figures and tables. The code and findings, targeted at both SatNOGS contributors and academic readers, will be fully referenced and open-sourced along with the project.

Sources: The background and project approach draw on SatNOGS documentation and community discussions[1][16][4][5][8][9][13][14], as well as general satellite communications principles. All cited material is from SatNOGS sources or public domain references.

---

[1] [7] DIY Satellite Ground Station and Network – SatNOGS

https://satnogs.org/diy-satellites/

[2] [3] [16] SatNOGS Network - Home

https://network.satnogs.org/

[4] SatNOGS DB - SatNOGS Wiki

https://wiki.satnogs.org/SatNOGS_DB

[5] API — SatNOGS DB 1.61+0.gcddf99e.dirty documentation

https://docs.satnogs.org/projects/satnogs-db/en/stable/api.html

[6] API — SatNOGS Network 1.113+0.g284910d.dirty documentation

https://docs.satnogs.org/projects/satnogs-network/en/stable/api.html

[8] [11] [12] Orbital pass - Wikipedia

https://en.wikipedia.org/wiki/Orbital_pass

[9] [10] Elevation Angle - Spire : Global Data and Analytics

https://spire.com/spirepedia/elevation-angle/

[13] [15] Operation - SatNOGS Wiki

https://wiki.satnogs.org/Operation

[14] Observations vetting - Need for a reform - Software - Libre Space Community

https://community.libre.space/t/observations-vetting-need-for-a-reform/3877