

HF Propagation Dashboard with Real-Time Validation

Overview

This system combines **theoretical propagation predictions** (ITU-R model) with **real-time validation** from PSKreporter to help you understand where your HF signal can actually reach.

Key Features

- ITU-R propagation predictions for all DXCC entities
- Real-time PSKreporter integration showing where your signal is being received
- Validation of predictions against actual reception reports
- DXCC tracking with worked/confirmed status
- 72-hour propagation forecasts
- Gray line visualization

Quick Start

1. Install Dependencies



bash

```
pip install requests --break-system-packages
```

2. Generate Base Predictions



bash

```
python generate_propagation.py
```

This creates:

- `propagation_data.json` - ITU-R predictions
- `dxcc_entities.json` - DXCC entity database
- `solar_data.json` - Current solar conditions

3. Add Real-Time Validation



bash

```
python generate_enhanced_predictions.py
```

This creates:

- enhanced_predictions.json - Predictions with PSKreporter validation
- pskreporter_data.json - Detailed reception reports

4. View Dashboard

Open dashboard_enhanced.html in your browser

Understanding Validation Status

✓ Confirmed (Green)

Propagation was **predicted** AND your signal is **being received**

- The model correctly predicted this path
- You have actual reception reports on PSKreporter
- **Action:** These bands/entities are confirmed workable right now

⚡ Unexpected (Blue)

Your signal is **being received** but was **NOT predicted**

- The model underestimated this path
- Actual propagation is better than predicted
- **Action:** Try these - conditions are better than expected!

? Unconfirmed (Gray)

Propagation was **predicted** but **no reception reports yet**

- The model says it should work
- No PSKreporter spots yet (might not be anyone listening)
- **Action:** Worth trying - prediction says it should work

Files

Python Scripts

- generate_propagation.py - Base ITU-R prediction generator
- pskreporter_api.py - PSKreporter API integration
- generate_enhanced_predictions.py - Combines predictions with PSKreporter
- test_proppy.py - Test script for proppy library (if available)

Dashboard Files

- dashboard_enhanced.html - Main dashboard with validation
- dashboard.html - Original dashboard (predictions only)

Data Files (Generated)

- propagation_data.json - Base predictions
- enhanced_predictions.json - Predictions + validation
- pskreporter_data.json - Detailed PSKreporter data
- dxcc_entities.json - DXCC database
- solar_data.json - Solar conditions
- qso_log.adi - Your ADIF log file (you provide)

Configuration

Edit the following in the Python scripts:



```
# Your callsign
CALLSIGN = "VE1ATM"

# Your location (Lunenburg, NS)
TX_LAT = 44.376
TX_LON = -64.317

# Your power
TX_POWER = 100 # Watts

# Your log file
LOG_FILE = Path('qso_log.adi')
```

Dashboard Tabs

Overview

- Solar conditions
- Live PSKreporter statistics
- Band-by-band success rates

Real-Time Validation

- **Confirmed:** Predictions validated by actual reception
- **Unexpected:** Better-than-predicted propagation
- **Unconfirmed:** Predictions not yet validated

Band Details

- Detailed predictions for each band

- MUF, signal strength, reliability
- Best times and angles

72-Hour Forecast

- Propagation predictions for next 3 days
- Hour-by-hour band conditions

DXCC Tracking

- Needed entities with propagation status
- Worked/confirmed status from your log
- Best bands for each needed entity

PSKreporter Integration

PSKreporter shows **actual stations receiving your signal**. The system:

1. Queries PSKreporter for your callsign
2. Gets reception reports from the last 60 minutes
3. Maps receivers to DXCC entities
4. Compares against theoretical predictions
5. Shows validation status for each entity

What the spots mean

When you see a spot on PSKreporter:

- **Receiver:** Station that decoded your signal
- **Grid:** Their Maidenhead grid square
- **SNR:** Signal-to-noise ratio (higher is better)
- **Band:** Frequency band of reception
- **Time:** When they heard you

About VOACAP

Current Status

The proppy package (v0.0.2) that you installed doesn't appear to be a complete VOACAP interface.

Professional VOACAP Options

1. **VOACAP Online** (www.voacap.com)
 - Web-based interface
 - No API available for automation
 - Manual predictions only
2. **pythonProp + voacapl**
 - Full VOACAP installation required
 - Linux-based GUI application
 - Complex setup but professional-grade
3. **ITU-R Model (Current)**
 - Good approximation
 - Validated against PSKreporter

- No installation complexity

Validation Against Reality

The beauty of the PSKreporter integration is that **you don't need perfect predictions** - you can see what's actually working!
The validation system tells you:

- Which predictions are accurate (confirmed)
- Where conditions are better than predicted (unexpected)
- Where to try based on predictions (unconfirmed)

Refresh Schedule

The system automatically refreshes:

- **Dashboard:** Every 5 minutes
- **Solar data:** On each prediction run
- **PSKreporter:** On each enhanced prediction run

Troubleshooting

No PSKreporter data

If you're not seeing any reception reports:

- Make sure you're transmitting (FT8, WSPR, etc.)
- Wait a few minutes for spots to appear
- Check PSKreporter website to verify your callsign appears
- Ensure you're using digital modes (PSK31, FT8, etc.)

Predictions seem wrong

The ITU-R model is an approximation. Use the PSKreporter validation to:

- Identify when the model is accurate
- Learn when to trust predictions
- Discover better-than-predicted opportunities

Dashboard not loading

1. Make sure you've run `generate_propagation.py` first
2. Check that JSON files exist in the same directory
3. Open browser console (F12) to see errors
4. Verify all files are in the same folder

Next Steps

1. Run the scripts and view the dashboard
2. Compare predictions vs actual PSKreporter spots
3. Note patterns in confirmed vs unexpected propagation
4. Use this data to optimize your operating schedule

Credits

- ITU-R propagation model implementation
 - PSKreporter API for real-time validation
 - Solar data from NOAA/SWPC
-

VE1ATM | Lunenburg, Nova Scotia | FN74ui