

Response to reviewers

September 19, 2025

We would like to thank the Reviewers for their constructive and thoughtful comments. We have addressed each point and revised the manuscript accordingly. Our detailed responses follow below.

———— Reviewer 1

———— Comment 1

The ‘Entoto Observatory’ can at best be called a ‘Variometer Station’ because it does not qualify as a magnetic observatory in the true sense of the word as no absolute measurements are being done to determine baseline values. In the manuscript the authors use both ‘station’ and ‘observatory’ which can confuse the reader of this paper

We thank the reviewer and agree that the station should not be referred to as an observatory. We now refer to it consistently as the Entoto Magnetometer Station and have clarified in the abstract and body that it is a variometer-only station hosted at the Entoto Observatory site, without absolute baseline measurements.

———— Comment 2

In line 7 the authors write ‘ENTOTO’, while in the title of the manuscript it is written ‘Entoto’. Consistency is advised.

We agree that consistency is important, and we have revised the manuscript accordingly to use the capitalisation “Entoto” throughout.

———— Comment 3

Figure 1 reveals that substantial magnetic gradients exist at the site of the station, reaching 240 nT/m in the North-West corner. The effect of this is that the area is not clean as a 10 nT/m gradient is normally the criterium for a clean site. It is recommended that the position of the station be shown in Figure 1 to give the reader a better appreciation of its location. These large gradients, indicative of magnetised rocks under the surface, can unfortunately lead to substantial induction effects on the magnetic field recordings, leading to larger than expected errors in the data. The authors should comment accordingly in the rewritten

paper.

We appreciate this observation. We have now added the station's precise position to Figure 1 and discussed the implications of high magnetic gradients. Although the gradient reaches 240 nT/m in the NW corner, the selected deployment site lies outside this region. As this is not an absolute observatory, we agree that such gradients could introduce induction effects but are acceptable given the station's purpose for space weather monitoring. A note on this limitation has been added to the discussion section.

Comment 4

In line 102 the authors mention the Keetmanshoop INTERMAGNET observatory. Please provide a reference. (Korte, M., M. Manda, H.-J. Linthe, A. Hemshorn, P. Kotzé and E. Ricaldi : New geomagnetic field observations in the South Atlantic Anomaly region. *Annals of Geophysics*, 52, 65-81, 2009.)

We thank the reviewer for this helpful reference suggestion. We have included the citation as recommended to support the mention of the Keetmanshoop INTERMAGNET observatory. The reference will be added to the bibliography and cited appropriately in the main text.

Comment 5

In line 106 the authors briefly mention the use of venting pipes for temperature stability. It is well-known that fluxgate magnetometers are extremely sensitive to temperature variations. Are venting pipes adequate to provide the required temperature stability? It would be informative for the reader to add a temperature variation plot to show that the environment inside the box is stable enough for the fluxgate sensor.

We thank the reviewer for raising this important point. We agree that venting pipes alone cannot guarantee the required thermal stability for fluxgate magnetometers. To assess and address this, we analysed the co-located sensor (T1) and electronics (T2) temperatures. As shown in Figure 3, the enclosure moderates the thermal environment, with modest daily variations (median $\sim 3.5^{\circ}\text{C}$ for T1 and $\sim 4.6^{\circ}\text{C}$ for T2; Table 1). This corresponds to an estimated median peak-to-peak effect of only ~ 2.6 nT on the H component.

To further mitigate residual effects, we derived and applied a temperature coefficient of $-0.733 \text{ nT } ^{\circ}\text{C}^{-1}$ for H. Figure 4 demonstrates that the correction slightly reduces the small diurnal variation in ΔF (on the order of ~ 2 nT on quiet days) and lowers its correlation with temperature. The effect is subtle, but consistent with expectations and shows that the correction acts as a safeguard. Overall, while the Entoto enclosure does not achieve observatory-grade stability, the combination of moderated thermal conditions and post-processing correction is adequate for variometer-grade operation.

Comment 6

What about lightning protection? The authors do not mention it in the paper, and is it of any concern?

At present, the Entoto Magnetometer Station does not have a dedicated lightning protection system installed. The sensor cable is not laid directly on the ground, which reduces the likelihood of induced currents reaching the sensor in the event of a lightning strike. The most vulnerable point is the AC power supply line, which could be affected by a direct hit; however, due to the isolated sensor grounding and layout, we do not expect significant risk of damage to the fluxgate sensor itself.

We have now noted in the revised manuscript that while no dedicated system is yet installed, lightning protection measures are under active consideration for future upgrades.

Comment 7

The authors mention that data are sampled at 1 min intervals using 1 sec values. How is this determined? Using an average over 60 sec, centred at the middle of each minute interval, or is it done by taking the average 30sec before and 30 sec after the minute? Please explain.

We thank the reviewer for the observation. The 1-minute values in our analysis are computed as centered means from the 1-second data, following the standard method recommended by IAGA and INTERMAGNET. We will clarify this in the manuscript accordingly.

Comment 8

Line 330: Please provide a website if no journal reference is available.

The citation in line 330 was an incomplete reference to the CHAOS-8.2 geomagnetic field model. We will correct this and include the full journal reference in the bibliography.

Reviewer 2

Comment 1

Line 48: Note INTERMAGNET does not operate observatories but rather enables exchange of data. The stations mentioned are (or were) operated by IPGP (AAE, MBO, SOK) and IPGP and CRAAG (TAM)

We thank the reviewer for this correction. We will revise the sentence to correctly attribute the operation of the observatories to IPGP and CRAAG, and clarify that INTERMAGNET facilitates global data exchange rather than operating observatories directly.

Comment 2

Figure 1. Error in caption. Longitude should read...

Corrected as suggested.

Comment 3

Figure 1 suggests high magnetic gradients (up to 20 nT/m) compared to the accepted recommendation of 1 nT/m for a typical geomagnetic observatory (Jankowski and Sucksdorff, 1996). However, that recommendation is typically applied to absolute magnetic observatories, so may not be significant if this observatory is primarily designed for space weather monitoring. Can this be commented on, particularly given the statement in Line 90 on static disturbances?

We now clarify that the 1 nT/m criterion (Jankowski and Sucksdorff, 1996) applies to absolute observatories. As Entoto is a variometer station intended for space weather research, higher gradients, while not ideal, are acceptable. This is now stated in both Figure 1 and the main text.

Comment 4

Line 75 The definition of L9 is made further on in the paper but can it be included here as 'lower K = 9 limit'?

We have moved the definition of L9 and now include "L9: the lower K = 9 threshold value" at its first mention in the text.

Comment 5

Line 58,84 Can these statements on the importance of an East African, equatorial station be strengthened by citation to literature on the requirement for regional space weather monitoring?

We have strengthened this with references to: - Hamid et al. (2014) - Myint et al. (2022) - Mungufeni et al. (2018)

These support the importance of equatorial stations for monitoring EEJ and regional space weather.

Comment 6

Line 113 Does the specification here of '1-second resolution' refer to direction or time? Could this be clarified?

Thank you for the observation. It refers to temporal resolution. We have updated the sentence to clarify this.

Comment 7

Line 124 The term 'geomagnetic coordinate system' would be more accurately described as 'geodetic coordinate system' or 'geographic coordinate system' given the definition of the XYZ co-ordinate system

We have corrected this to "geographic coordinate system".

Comment 8

Line 131 Is the fact that the data are sampled at one-minute intervals contradictory to the 1-second resolution referred to in Line 124? Are the data down sampled or filtered to one-minute? If so, can this process be defined i.e. is a specific filter used?

We clarified that no digital filtering is applied; 1-minute values are computed as simple centered means of 60 consecutive 1-second samples.

————— Comment 9

Line 172 What are the sources of non-geomagnetic noise in this frequency band (2-minute period) and can these be filtered without attenuating the signal of interest in the same band?

Non-geomagnetic noise sources include temperature drift and anthropogenic EM interference. While some are corrected (e.g., temperature), others cannot be removed without potentially attenuating ionospheric signals. We have added a note on these trade-offs.

————— Comment 10

Figures 9 and 10 The fitting of the mean daily maximum EEJ amplitude in the figures is close to the median daily maximum. If the daily maximum is normally distributed for both quiet and storm conditions, then the mean and the median will, of course, be equal but can it be confirmed that the plots show the mean and not the median?

These values are confirmed to be the mean daily maxima, as implemented in our processing scripts (Appendix).

————— Comment 11

Given the noted operational difficulties in maintaining long-term magnetic stations in the region, can the authors comment further on measures taken to ensure the ENTOTO station will continue to operate in the long-term? For example, are there formal long-term agreements in place between the institutes collaborating on this project?

Thank you for raising this important point. We have added a note in the manuscript outlining the sustainability strategy for the Entoto station.

The deployment is supported by a formal Memorandum of Understanding (MoU) between the South African National Space Agency (SANSA) and the Space Science and Geospatial Institute (SSGI) in Ethiopia. This framework ensures long-term institutional commitment, while also facilitating data sharing, infrastructure access, and joint research efforts.

We would like to acknowledge and credit earlier initiatives, particularly the efforts of teams such as the AMBER group, whose deployments across Africa laid critical groundwork. From their and other teams' experiences, we've learned the importance of maintaining continuous and open communication between hosting and supporting institutes. This includes regular

updates between technicians, researchers, and software teams to avoid the “black box” problem often encountered in remote or distributed sensor networks.

Our approach prioritises mutual scientific partnership, with cross-training on instrument maintenance, data processing, and analysis. The aim is to support the SSGI to one day be the second geomagnetic working group on the continent, who in turn can support SANSA in its endeavours. This ensures operational resilience and builds local expertise and longevity of instrument networks. We also continue to pursue travel and capacity-building grants to support technical exchanges and joint workshops, reinforcing both technical continuity and regional collaboration.

Appendix: EEJDSTv4.py, Python Script for EEJ Signal Processing

```
1 import os
2 import pandas as pd
3 import numpy as np
4 import chaosmagpy as cp
5 import requests
6 import matplotlib.pyplot as plt
7 from datetime import datetime, timedelta
8 from sklearn.linear_model import LinearRegression
9 import importlib
10 import calcChaos
11 importlib.reload(calcChaos)
12 from calcChaos import chaos, chaos_ext, datetime_to_decimal_year
13 import re
14 from datetime import datetime
15 from tqdm import tqdm
16 import matplotlib.dates as mdates
17
18 def load_entoto_data(directory):
19     all_data = []
20
21     def extract_date(filename):
22         match = re.search(r'ent(\d{8})pmin\.min', filename)
23         if match:
24             return datetime.strptime(match.group(1), '%Y%m%d')
25         return datetime.min # fallback
26
27     # List and sort .min files by date in filename
28     min_files = sorted(
29         [f for f in os.listdir(directory) if f.endswith('.min')],
30         key=extract_date
31     )
32
33     # Progress bar over files
34     for file in tqdm(min_files, desc="Loading Entoto.min files"):
35         file_path = os.path.join(directory, file)
36         try:
37             df = pd.read_csv(
38                 file_path,
39                 sep=r'\s+',
40                 comment='#',
41                 header=None,
```

```

42         skiprows=16,
43         names=["DATE", "TIME", "DOY", "ENTX", "ENTY", "ENTZ", "ENTF"],
44         engine='python',
45         on_bad_lines='skip'
46     )
47     df.replace(99999.0, np.nan, inplace=True)
48     # Combine DATE and TIME into a single DATETIME column
49     df['DATETIME'] = pd.to_datetime(df['DATE'] + '_' + df['TIME'],
50                                     errors='coerce')
51
52     # Drop the original separate DATE and TIME columns
53     df.drop(columns=['DATE', 'TIME'], inplace=True)
54
55     all_data.append(df)
56 except Exception as e:
57     print(f"Error processing {file}: {e}")
58
59 return pd.concat(all_data, ignore_index=True) if all_data else None
60
61 # Step 2: Calculate H component from X and Y
62 def calculate_H_component(df):
63     df['H'] = np.sqrt(df['ENTX']**2 + df['ENTY']**2)
64     return df
65
66 # Step 3: Remove CHAOS internal field to get H_residual
67 def remove_internal_field(df, station_lat=9.108, station_lon=38.807,
68                             station_alt=2450):
69     df['H_internal'] = np.nan
70     df['H_residual'] = np.nan
71
72     # Group data by date
73     df['DATE'] = df['DATETIME'].dt.date
74     unique_dates = df['DATE'].unique()
75
76     for date in unique_dates:
77         daily_df = df[df['DATE'] == date]
78         pkl_filename = f"internal_field_{date}.pkl"
79
80         if os.path.exists(pkl_filename):
81             daily_internal = pd.read_pickle(pkl_filename)
82         else:
83             # Compute internal field for each timestamp
84             internal_values = []
85             for dt in daily_df['DATETIME']:
86                 Bx, By, Bz = chaos(datetime_to_decimal_year(dt), station_lat,
87                                     station_lon, station_alt)
88                 H_internal = np.sqrt(Bx**2 + By**2)
89                 internal_values.append(H_internal)
90             daily_internal = pd.Series(internal_values, index=daily_df.index)
91             daily_internal.to_pickle(pkl_filename)
92
93     df.loc[daily_df.index, 'H_internal'] = daily_internal
94     df.loc[daily_df.index, 'H_residual'] = df.loc[daily_df.index, 'H'] -
95         daily_internal

```

```

95     df.drop(columns=['DATE'], inplace=True)
96     return df
97
98 def compute_external_field(df, station_lat=9.108, station_lon=38.807,
99     station_alt=2450):
100     df['H_external'] = np.nan
101
102     # Group data by date
103     df['DATE'] = df['DATETIME'].dt.date
104     unique_dates = df['DATE'].unique()
105
106     for date in unique_dates:
107         daily_df = df[df['DATE'] == date]
108         pkl_filename = f"external_field_{date}.pkl"
109
110         if os.path.exists(pkl_filename):
111             daily_external = pd.read_pickle(pkl_filename)
112         else:
113             # Compute external field for each timestamp
114             external_values = []
115             for dt in daily_df['DATETIME']:
116                 Bx_ext, By_ext, Bz_ext = chaos_ext(datetime_to_decimal_year(dt),
117                     station_lat, station_lon, station_alt)
118                 H_external = np.sqrt(Bx_ext**2 + By_ext**2)
119                 external_values.append(H_external)
120             daily_external = pd.Series(external_values, index=daily_df.index)
121             daily_external.to_pickle(pkl_filename)
122
123         df.loc[daily_df.index, 'H_external'] = daily_external
124
125     df.drop(columns=['DATE'], inplace=True)
126     return df
127
128 # Step 4: Estimate average night-time magnetospheric field from H_residual
129 def estimate_magnetospheric_component(df):
130     night_mask = (df['DATETIME'].dt.hour >= 18) | (df['DATETIME'].dt.hour < 6)
131     df['H_magnetospheric'] = df.loc[night_mask, 'H_residual']
132     return df
133
134 # Step 5: Extract daytime EEJ signal from H_residual (still contains
135 # magnetospheric field)
136 def extract_eej_signal(df):
137     day_mask = (df['DATETIME'].dt.hour >= 9) & (df['DATETIME'].dt.hour <= 15)
138     df['EEJ'] = df.loc[day_mask, 'H_residual']
139     return df
140
141 def fetch_dst_index(start_date, end_date):
142     dst_records = []
143     filepath="/home/amore/Documents/00Data/Dst_oct2024_apr2025.dat"
144     with open(filepath, 'r') as file:
145         for line in file:
146             parts = line.strip().split()
147             if len(parts) < 26:
148                 continue # Skip malformed lines

```

```

149         # Parse date from ID like DST2410*01PPX120
150         id_str = parts[0]
151         year = int("20" + id_str[3:5])
152         month = int(id_str[5:7])
153         day = int(id_str.split("*")[1][:2])
154
155         try:
156             hourly_values = [int(val) for val in parts[2:26]] # Skip 2nd
157                             # column (always 0), then 24 values
158         except ValueError:
159             continue # Skip lines with invalid integer entries
160
161         for hour, dst in enumerate(hourly_values):
162             dt = datetime(year, month, day, hour)
163             if start_date <= dt <= end_date:
164                 dst_records.append({'DATETIME': dt, 'Dst': dst})
165
166         return pd.DataFrame(dst_records)
167
168 # Step 7: Perform linear regression between Dst and H_residual to estimate
169 # magnetospheric field
170 # Step 8: Subtract modeled magnetospheric contribution to get cleaned EEJ
171 # signal (HEEJ)
172
173 def perform_linear_regression(df, dst_data):
174     # Ensure both are sorted by time
175     df = df.sort_values('DATETIME')
176     dst_data = dst_data.sort_values('DATETIME')
177
178     # Merge with nearest previous Dst value (i.e., forward fill)
179     merged = pd.merge_asof(df, dst_data[['DATETIME', 'Dst']], on='DATETIME',
180                             direction='backward')
181
182     # Drop rows with missing data
183     merged_clean = merged.dropna(subset=['Dst', 'H_residual'])
184
185     # Prepare regression inputs
186     x = merged_clean['Dst'].values.reshape(-1, 1)
187     y = merged_clean['H_residual'].values.reshape(-1, 1)
188
189     # Perform linear regression
190     reg = LinearRegression().fit(x, y)
191     merged_clean['Hmag_model'] = reg.predict(x)
192
193     # Merge the modeled magnetospheric signal back into the full dataset
194     merged = pd.merge(merged, merged_clean[['DATETIME', 'Hmag_model']], on='
195                       DATETIME', how='left')
196
197     # Compute the cleaned EEJ signal
198     merged['HEEJ'] = merged['H_residual'] - merged['Hmag_model']
199
200     return merged, reg.coef_[0][0]
201
202 # After regression, keep only daytime values of the cleaned signal:
203 def extract_daytime_eej(df):
204     df = df.copy() # prevent SettingWithCopyWarning

```

```

201 df['HEEJ_daytime'] = np.nan # initialize the column with NaNs
202 day_mask = (df['DATETIME'].dt.hour >= 9) & (df['DATETIME'].dt.hour <= 15)
203 df.loc[day_mask, 'HEEJ_daytime'] = df.loc[day_mask, 'HEEJ']
204 return df
205
206
207 def plot_monthly_magnetospheric_vs_dst(df, dst_data):
208     df['YEAR'] = df['DATETIME'].dt.year
209     df['MONTH'] = df['DATETIME'].dt.month
210
211     for (year, month), group in df.groupby(['YEAR', 'MONTH']):
212         start = group['DATETIME'].min()
213         end = group['DATETIME'].max()
214         dst_subset = dst_data[(dst_data['DATETIME'] >= start) & (dst_data['
                DATETIME'] <= end)]
215
216         plt.figure(figsize=(12, 6))
217         ax1 = plt.gca()
218         ax1.plot(group['DATETIME'], group['H_magnetospheric'], label='
                H_magnetospheric', color='blue')
219         ax1.set_ylabel('H_magnetospheric (nT)', color='blue')
220         ax1.tick_params(axis='y', labelcolor='blue')
221
222         ax2 = ax1.twinx()
223         ax2.plot(dst_subset['DATETIME'], dst_subset['Dst'], label='Dst Index',
                color='red')
224         ax2.set_ylabel('Dst Index (nT)', color='red')
225         ax2.tick_params(axis='y', labelcolor='red')
226
227         plt.title(f'Magnetospheric Signal vs Dst - {year}-{month:02}')
228         ax1.set_xlabel('Date')
229         plt.grid()
230         plt.tight_layout()
231         plt.savefig(f'Magnetospheric_vs_Dst_{year}_{month:02}.png', dpi=300)
232         plt.close()
233
234
235
236 # Step 10: Plot raw EEJ signal (before Dst correction)
237 def plot_superposed_epoch_eej(df):
238     if 'EEJ' not in df.columns:
239         print("EEJ not available. Skipping superposed epoch plots.")
240         return
241
242     df['YEAR'] = df['DATETIME'].dt.year
243     df['MONTH'] = df['DATETIME'].dt.month
244     df['HOUR_MIN'] = df['DATETIME'].dt.strftime('%H:%M')
245
246     # Keep only daytime
247     df_daytime = df[(df['DATETIME'].dt.hour >= 9) & (df['DATETIME'].dt.hour <=
        15)].copy()
248
249     # Round time to 30-minute bins
250     df_daytime['DATETIME'] = df_daytime['DATETIME'].dt.floor('30T')
251     df_daytime['HOUR_MIN'] = df_daytime['DATETIME'].dt.strftime('%H:%M')
252
253     for (year, month), group in df_daytime.groupby(['YEAR', 'MONTH']):

```

```

254     # Pivot: time of day (rows)      day (columns)
255     pivot = group.pivot_table(index='HOUR_MIN', columns=group['DATETIME'].
        dt.date, values='EEJ')
256
257     # Compute mean and std at each time bin
258     mean_series = pivot.mean(axis=1)
259     std_series = pivot.std(axis=1)
260
261     # Convert HOUR_MIN back to datetime-like index for proper plotting
262     time_labels = [datetime.strptime(t, '%H:%M') for t in mean_series.index
        ]
263
264     plt.figure(figsize=(10, 5))
265     plt.plot(time_labels, mean_series, label='Mean_EEJ', color='orange')
266     plt.fill_between(time_labels, mean_series - std_series, mean_series +
        std_series,
267                     color='orange', alpha=0.3, label=' 1 Std_Dev')
268
269     # Format x-axis to show only time (HH:MM)
270     import matplotlib.dates as mdates
271     plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
272
273     plt.xlabel('Local_Time_(24-hour)')
274     plt.ylabel('EEJ_Magnetic_Field_(nT)')
275     plt.title(f'Superposed_Epoch_of_EEJ_{year}-{month:02}')
276     plt.grid()
277     plt.legend()
278     filename = f'Superposed_EEJ_{year}-{month:02}.png'
279     plt.savefig(filename, dpi=300)
280     plt.close()
281
282
283
284 def plot_superposed_epoch_eej_vs_heel(df):
285     if 'EEJ' not in df.columns or 'HEEJ' not in df.columns:
286         print("EEJ or HEEJ not available. Skipping superposed comparison plots.
            ")
287         return
288
289     df['YEAR'] = df['DATETIME'].dt.year
290     df['MONTH'] = df['DATETIME'].dt.month
291     df['HOUR_MIN'] = df['DATETIME'].dt.strftime('%H:%M')
292
293     # Filter for daytime hours
294     df_daytime = df[(df['DATETIME'].dt.hour >= 9) & (df['DATETIME'].dt.hour <=
        15)].copy()
295     df_daytime['DATETIME'] = df_daytime['DATETIME'].dt.floor('30min')
296     df_daytime['HOUR_MIN'] = df_daytime['DATETIME'].dt.strftime('%H:%M')
297
298     for (year, month), group in df_daytime.groupby(['YEAR', 'MONTH']):
299         # Pivot tables for EEJ and HEEJ
300         pivot_eej = group.pivot_table(index='HOUR_MIN', columns=group['DATETIME
            '].dt.date, values='EEJ')
301         pivot_heel = group.pivot_table(index='HOUR_MIN', columns=group['
            DATETIME'].dt.date, values='HEEJ')
302
303     # Compute mean values

```

```

304     mean_eej = pivot_eej.mean(axis=1)
305     mean_heej = pivot_heej.mean(axis=1)
306
307     # Create time axis
308     time_labels = [datetime.strptime(t, '%H:%M') for t in mean_eej.index]
309
310     plt.figure(figsize=(10, 5))
311     plt.plot(time_labels, mean_eej, label='Raw_EEJ', color='orange')
312     plt.plot(time_labels, mean_heej, label='Dst-corrected_EEJ_(HEEJ)',
313             color='green')
314
315     plt.gca().axis.set_major_formatter(mdates.DateFormatter('%H:%M'))
316     plt.xlabel('Local_Time_(24-hour)')
317     plt.ylabel('Magnetic_Field_(nT)')
318     plt.title(f'Superposed_Epoch_of_EEJ_vs_HEEJ_{year}-{month:02}')
319     plt.grid()
320     plt.legend()
321     plt.tight_layout()
322     filename = f'Superposed_EEJ_vs_HEEJ_{year}-{month:02}.png'
323     plt.savefig(filename, dpi=300)
324     plt.close()
325
326
327
328 def filter_by_month(df, year, month):
329     # Ensure DATETIME is datetime type
330     df['DATETIME'] = pd.to_datetime(df['DATETIME'])
331
332     # Filter for the specific year and month
333     filtered_df = df[(df['DATETIME'].dt.year == year) & (df['DATETIME'].dt.
334         month == month)]
335
336     # Filter for times between 09:00 and 15:00
337     filtered_df = filtered_df[(filtered_df['DATETIME'].dt.hour >= 9) & (
338         filtered_df['DATETIME'].dt.hour < 15)]
339
340     print(filtered_df)
341
342 # Main execution pipeline
343 def main():
344     directory = '/home/amore/Documents/00Data/ENTO'
345     output_file = 'testfile.pkl'
346
347     df = load_entoto_data(directory)
348     if df is None:
349         print("No_data_files_found.")
350         return
351
352     df = calculate_H_component(df)
353
354     df = remove_internal_field(df)
355
356     df = compute_external_field(df)
357
358     df = extract_eej_signal(df)

```

```

358
359     df = estimate_magnetospheric_component(df)
360
361     plot_superposed_epoch_eej(df)
362
363     start_date, end_date = df['DATETIME'].min(), df['DATETIME'].max()
364     dst_data = fetch_dst_index(start_date, end_date)
365
366     if dst_data is not None and not dst_data.empty:
367         df, k = perform_linear_regression(df, dst_data)
368         print(f"Estimated scaling factor k: {k:.3f}")
369         df = extract_daytime_eej(df)
370         filter_by_month(df, 2025, 3)
371         plot_monthly_magnetospheric_vs_dst(df, dst_data)
372         plot_superposed_epoch_eej_vs_heel(df)
373
374     else:
375         print("Warning: No Dst data available for the date range. Skipping regression and EEJ correction plot.")
376
377     df.to_pickle(output_file)
378     print(f"Processed data saved to {output_file}")
379

```

Listing 1: Python script used to compute and aggregate EEJ amplitude from Entoto station data.