

# SafeNet: SwArm for Earthquake Perturbations Identification Using Deep Learning Networks

Pan Xiong <sup>1</sup>, Dedalo Marchetti <sup>2,3</sup>, Angelo De Santis <sup>3</sup>, Xuemin Zhang <sup>1</sup> and Xuhui Shen <sup>4,\*</sup>

<sup>1</sup> Institute of Earthquake Forecasting, China Earthquake Administration, Beijing 100036, China; xiongpan@ief.ac.cn (P.X.); zxm@ief.ac.cn (X.Z.)

<sup>2</sup> College of Instrumentation and Electrical Engineering, Jilin University, Changchun 130061, China; dedalomarchetti@jlu.edu.cn

<sup>3</sup> Istituto Nazionale di Geofisica e Vulcanologia, Via di Vigna Murata 605, 00143 Rome, Italy; angelo.desantis@ingv.it

<sup>4</sup> National Institute of Natural Hazards, Ministry of Emergency Management of China, Beijing, 100085, China

\* Correspondence: shenxh@seis.ac.cn

## Contents of this file

Text S1. Swarm Alpha night-time tracks before M7.9 Sumatra 02-03-2016 earthquake.

Text S2. Swarm Alpha night-time tracks before M7.8 Ecuador 16-04-2016 earthquake.

Text S3. Performance metrics.

Text S4. The SafeNet model structure.

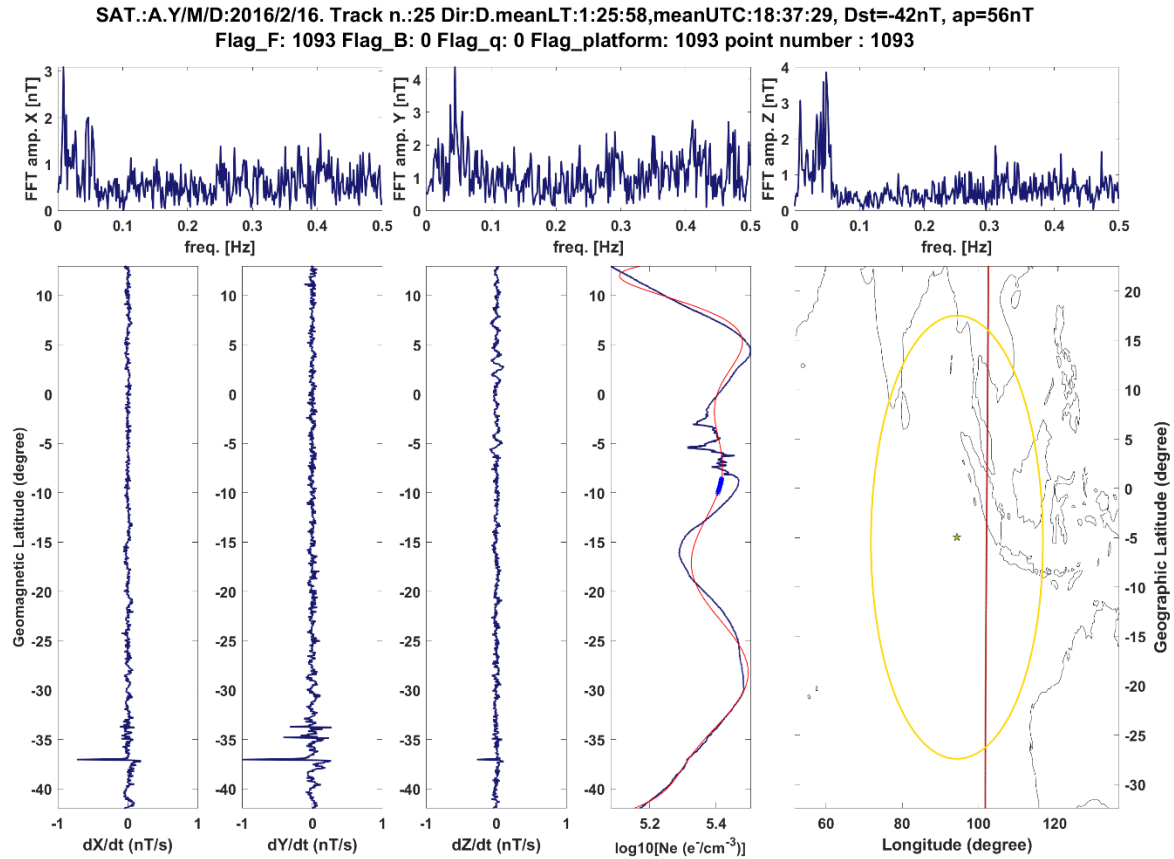
Figure S25. The flowchart of the proposed deep learning framework.

Table S1. Search space of parameters for the SafeNet model.

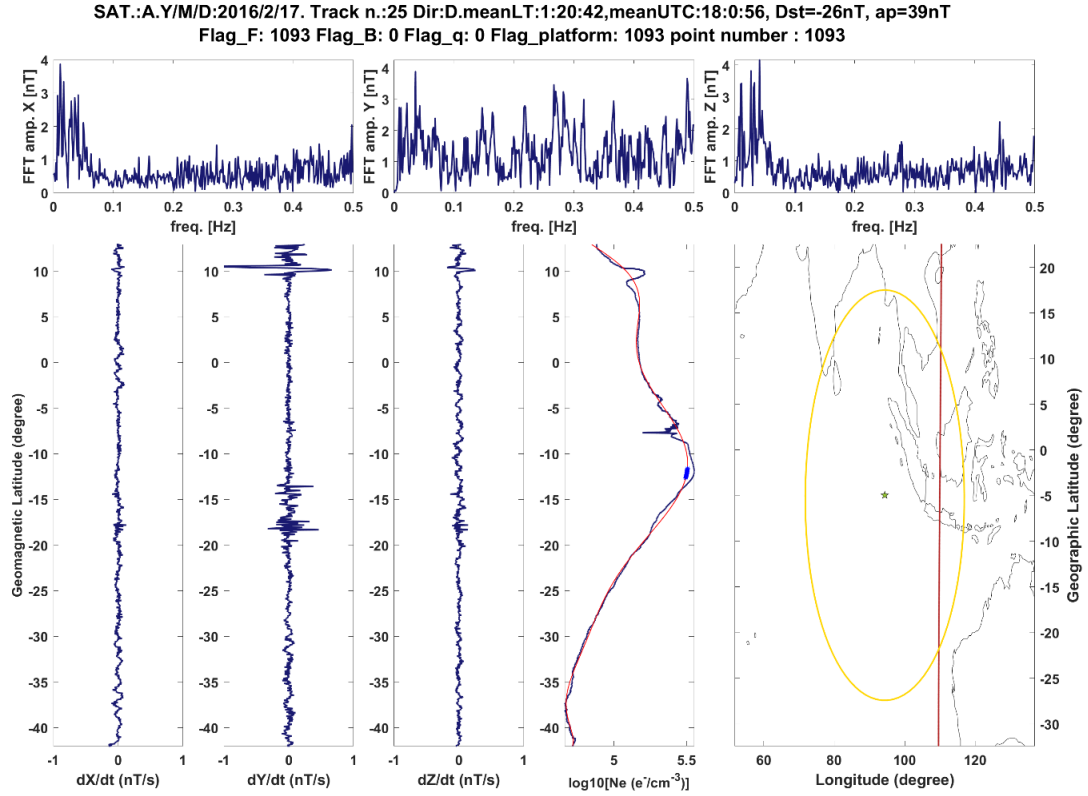
# Text S1. Swarm Alpha night-time tracks before M7.9 Sumatra 02-03-2016 earthquake.

In the following pictures (from Figure S1 to Figure S15) we show one Swarm Alpha night time track that crossed Dobrovoslky's area for each day from 16 February 2016 until 1 March 2016, i.e. the day before the occurrence of Mw=7.9 Sumatra 2016 earthquake at 12:49:48 UT. The day of the earthquake the Swarm satellite passed in nighttime across the Dobrovoslky's area about 5 hours after the seismic event and so it has not been shown here.

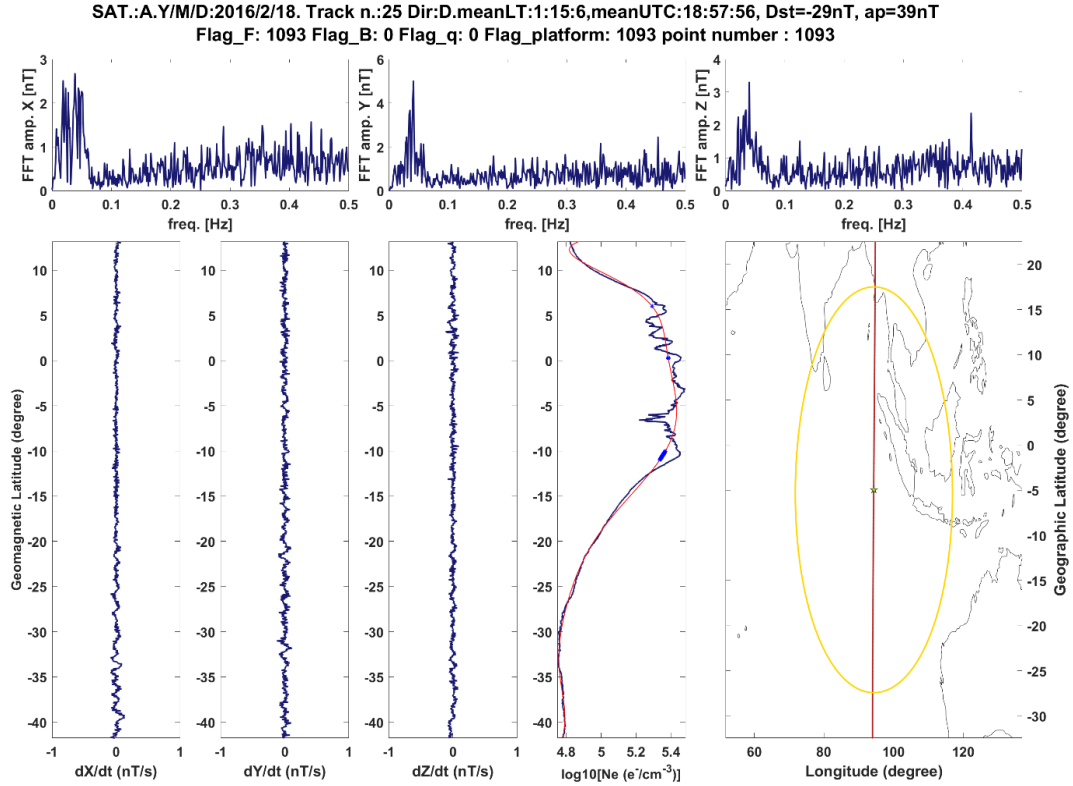
It is possible to note that with disturbed geomagnetic conditions not always happens anomalies, as in Figure S3. What about the same condition the Figure S2 shown an anomaly aligned with the plate boundaries, but due to the geomagnetic activity it is difficult to take any conclusion and future studies are necessary.



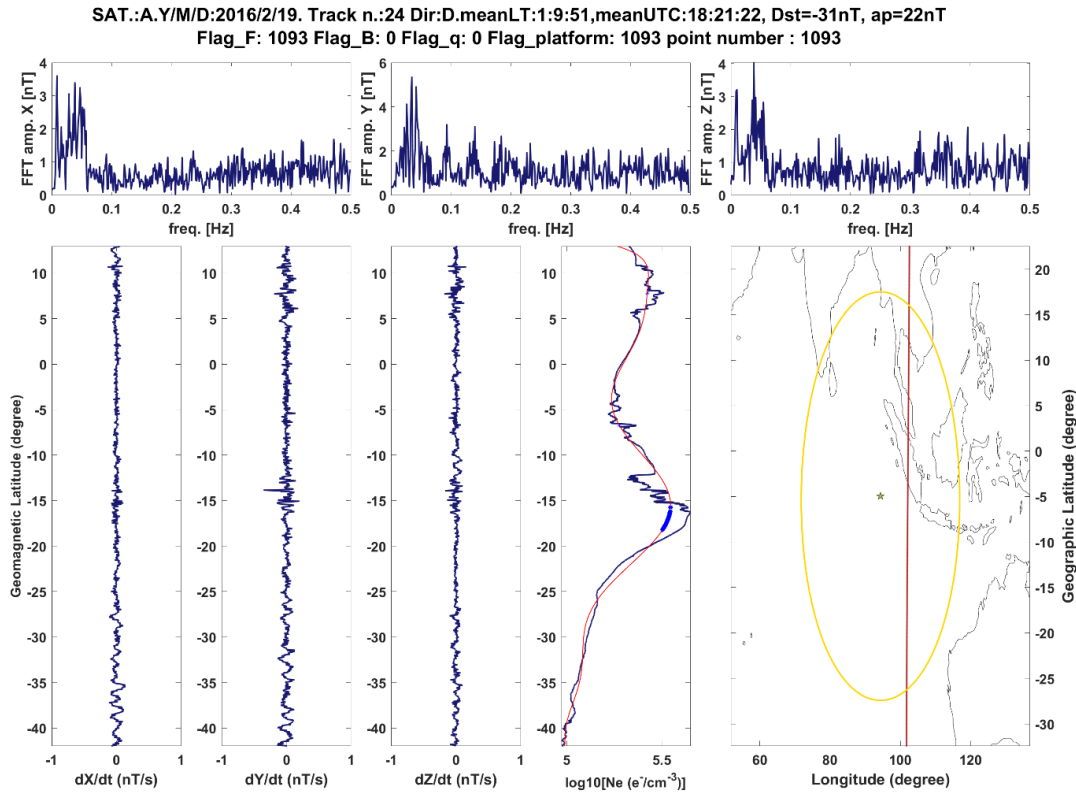
**Figure S1.** Swarm Alpha satellite nighttime track 25 of 2016-02-16 acquired 15 days before the M7.9 Sumatra 2016 earthquake.



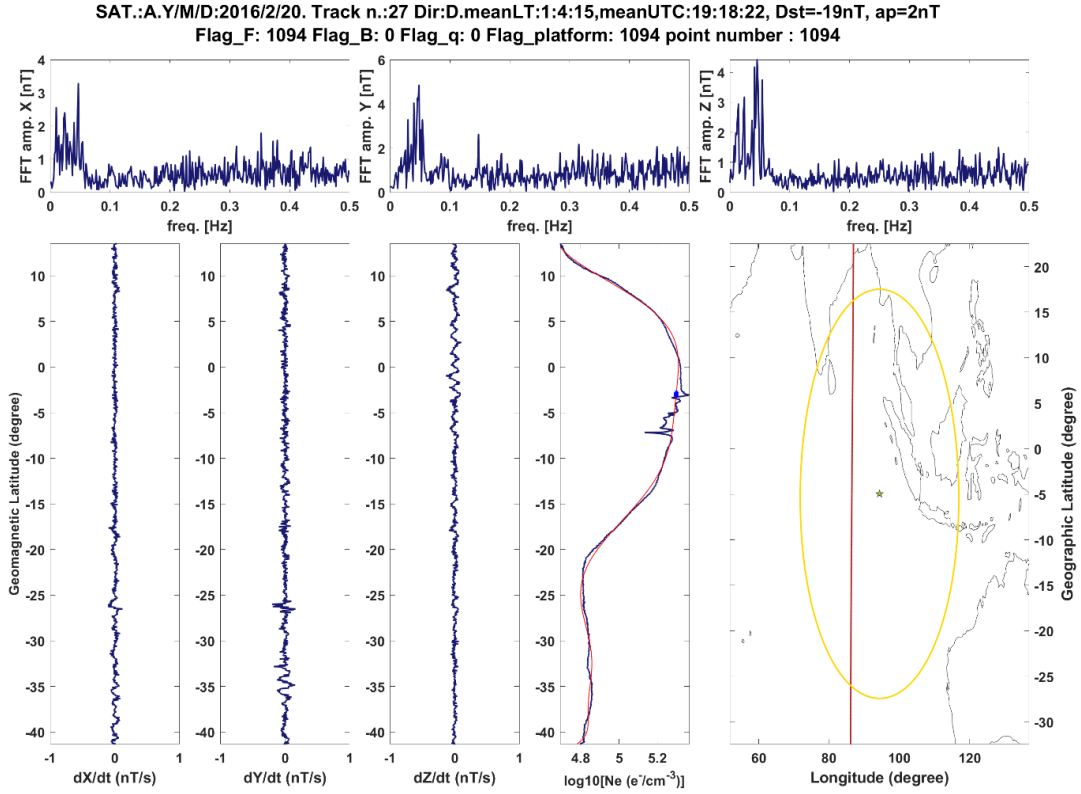
**Figure S2.** Swarm Alpha satellite nighttime track 25 of 2016-02-17 acquired 14 days before the M7.9 Sumatra 2016 earthquake.



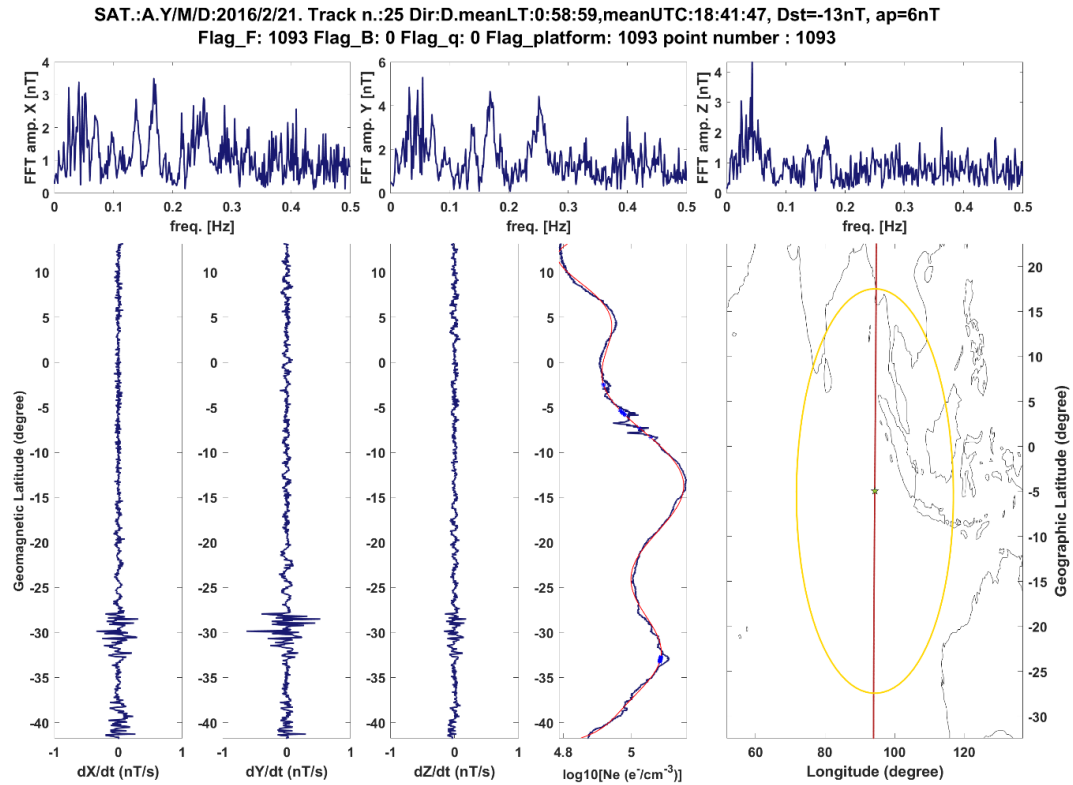
**Figure S3.** Swarm Alpha satellite nighttime track 25 of 2016-02-18 acquired 13 days before the M7.9 Sumatra 2016 earthquake.



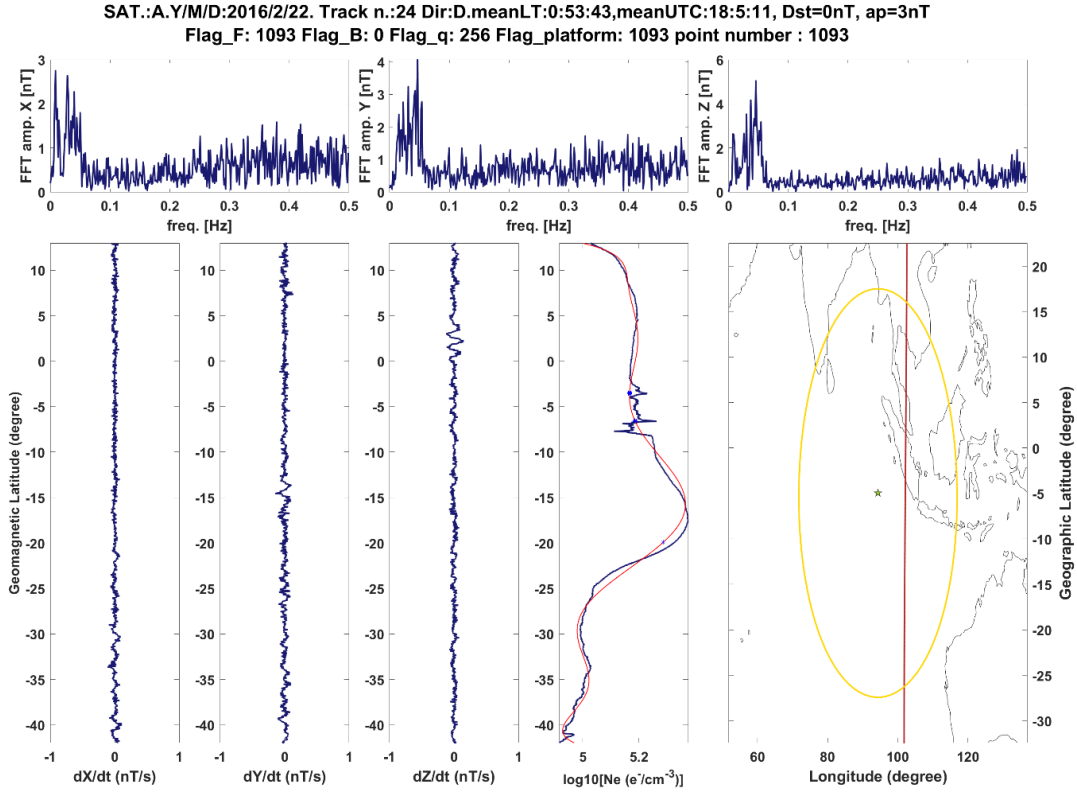
**Figure S4.** Swarm Alpha satellite nighttime track 24 of 2016-02-19 acquired 12 days before the M7.9 Sumatra 2016 earthquake.



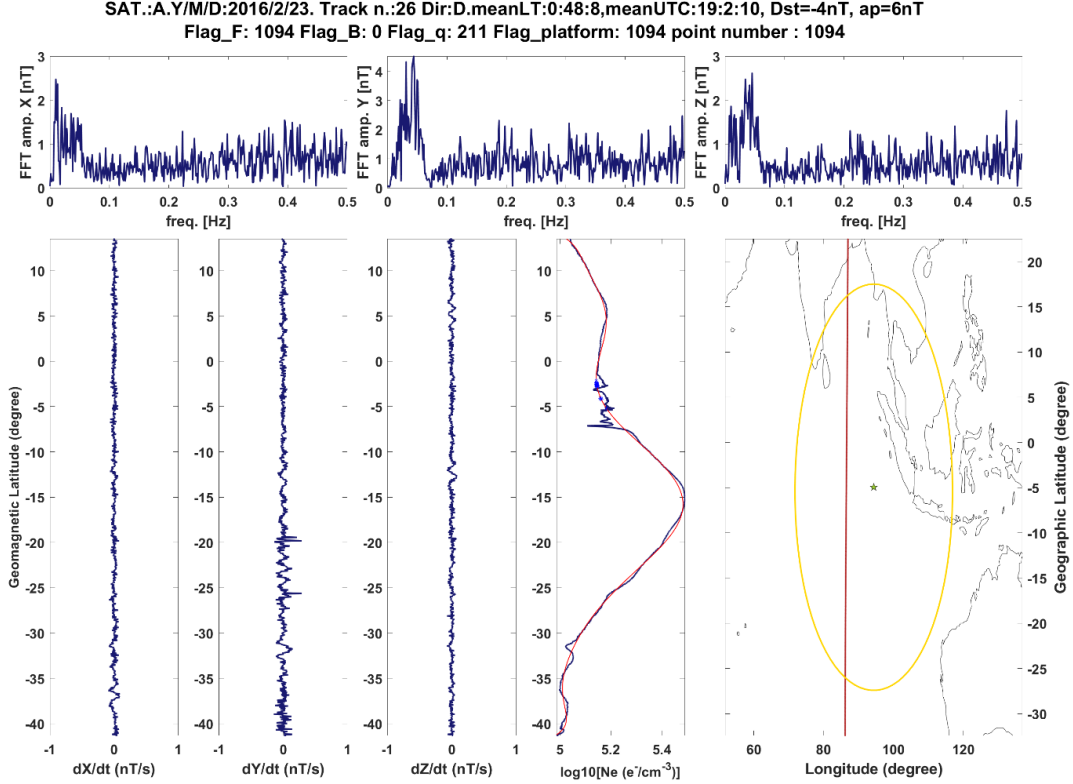
**Figure S5.** Swarm Alpha satellite nighttime track 27 of 2016-02-20 acquired 11 days before the M7.9 Sumatra 2016 earthquake.



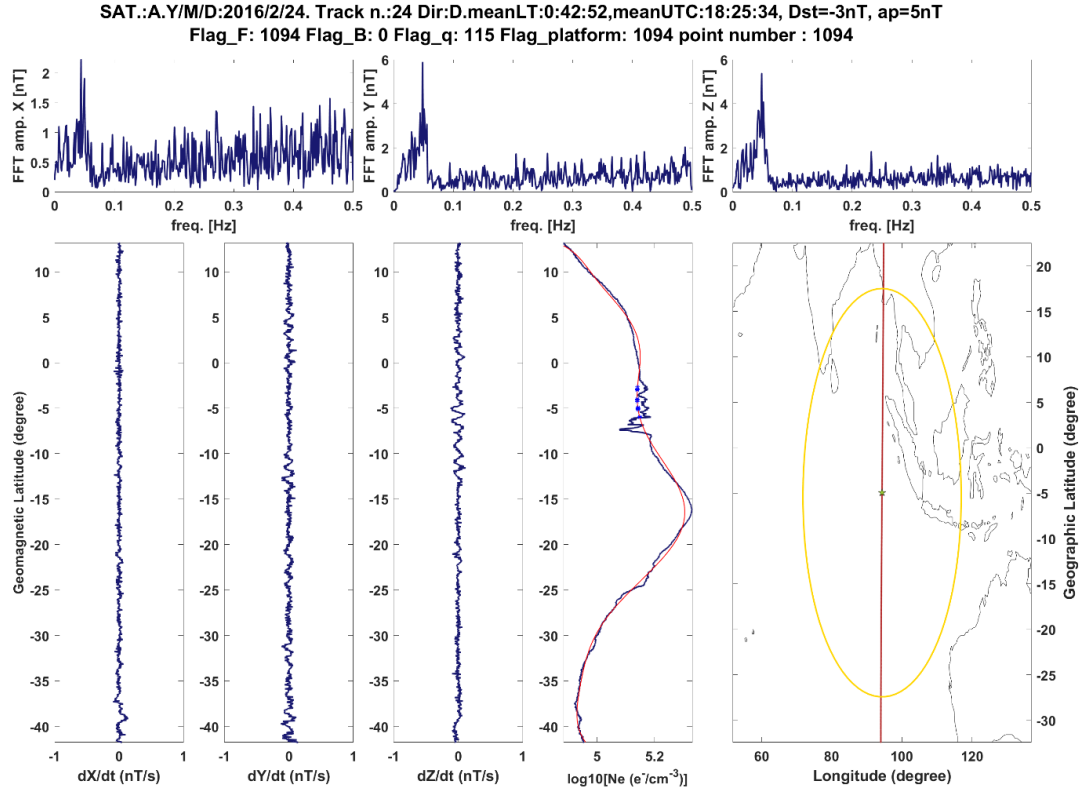
**Figure S6.** Swarm Alpha satellite nighttime track 25 of 2016-02-21 acquired 10 days before the M7.9 Sumatra 2016 earthquake.



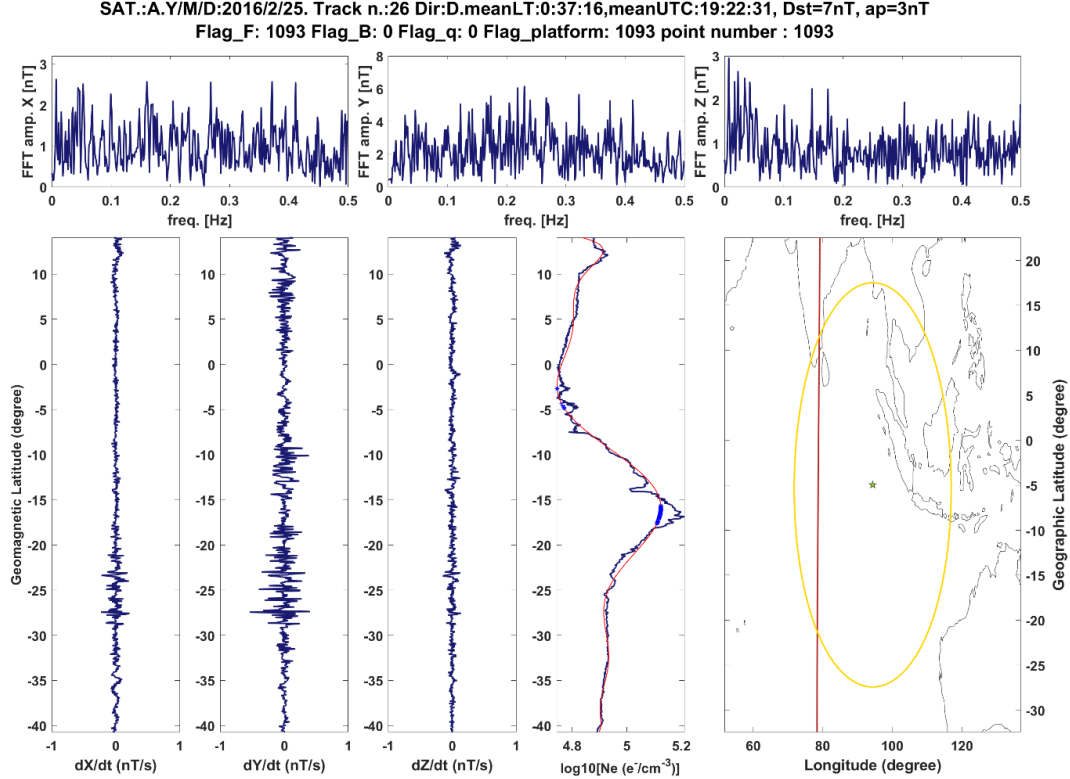
**Figure S7.** Swarm Alpha satellite nighttime track 24 of 2016-02-22 acquired 9 days before the M7.9 Sumatra 2016 earthquake.



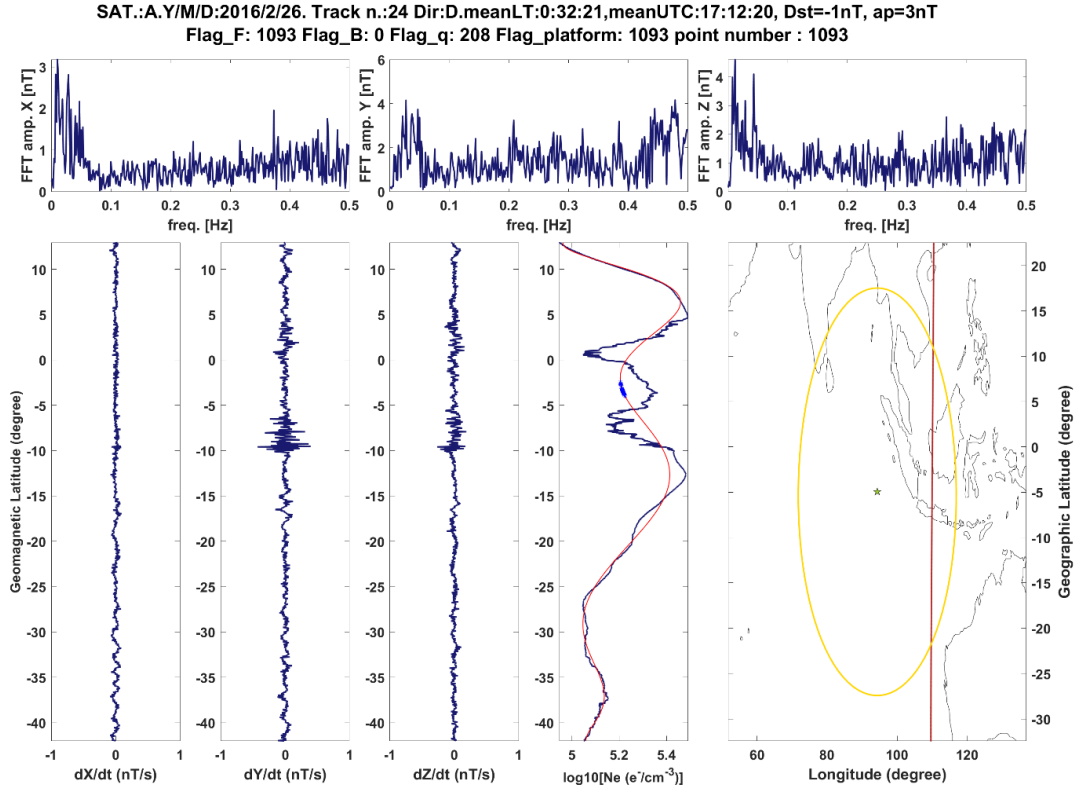
**Figure S8.** Swarm Alpha satellite nighttime track 26 of 2016-02-26 acquired 8 days before the M7.9 Sumatra 2016 earthquake.



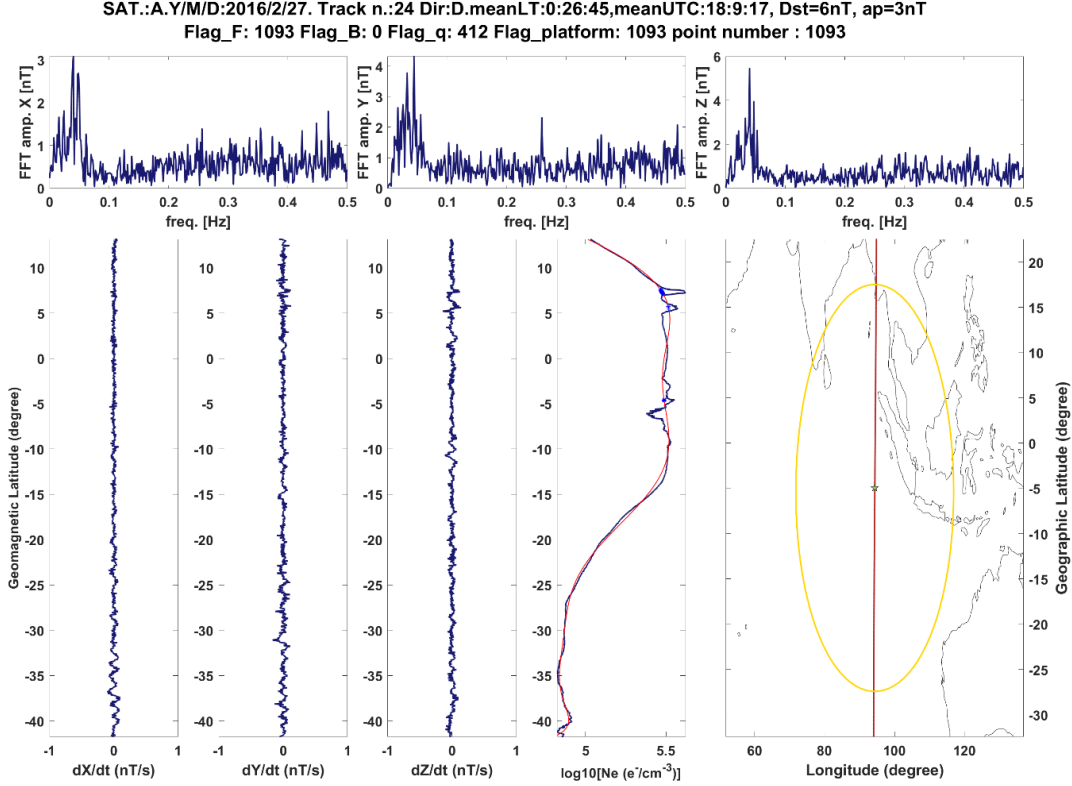
**Figure S9.** Swarm Alpha satellite nighttime track 24 of 2016-02-24 acquired 7 days before the M7.9 Sumatra 2016 earthquake.



**Figure S10.** Swarm Alpha satellite nighttime track 26 of 2016-02-25 acquired 6 days before the M7.9 Sumatra 2016 earthquake.

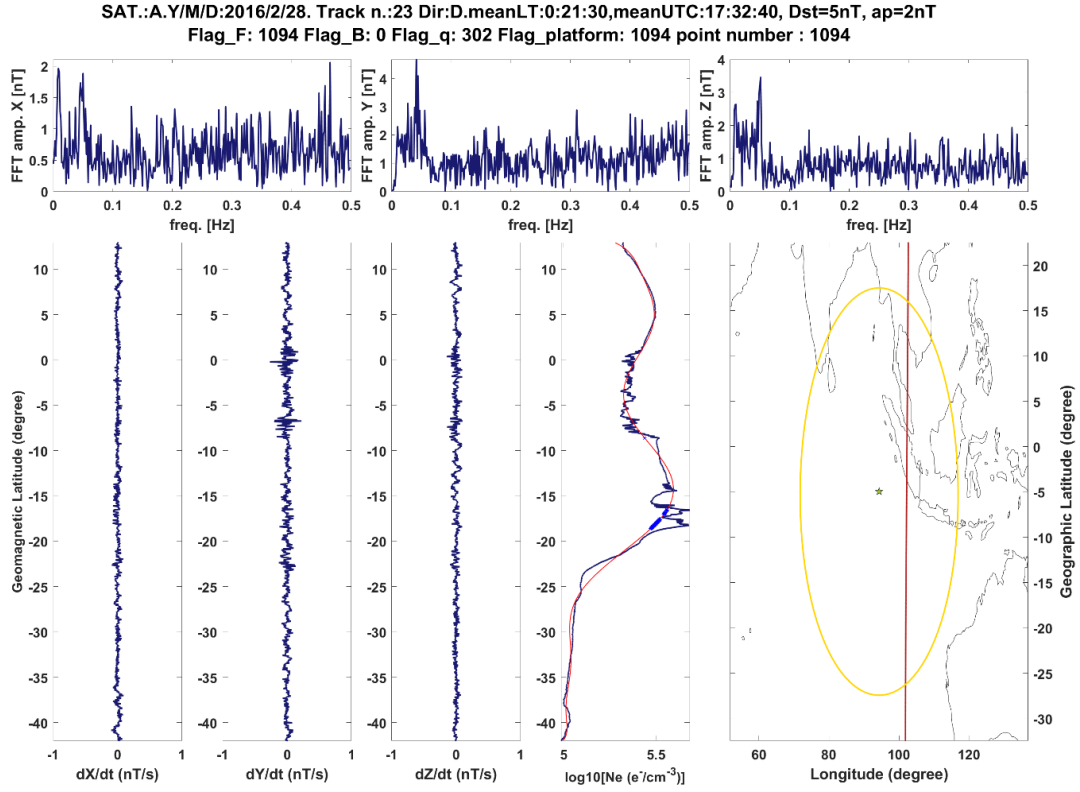


**Figure S11.** Swarm Alpha satellite nighttime track 24 of 2016-02-26 acquired 5 days before the M7.9 Sumatra 2016 earthquake.

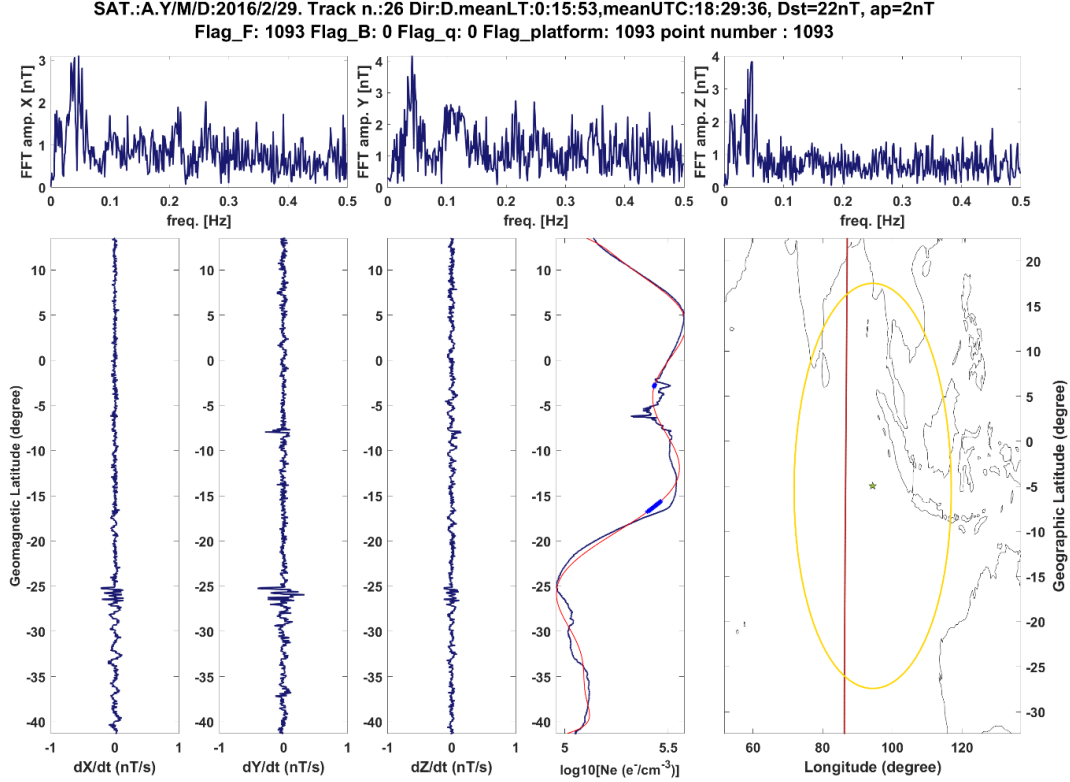


**Figure S12.** Swarm Alpha satellite nighttime track 24 of 2016-02-27 acquired 4 days before the M7.9 Sumatra 2016 earthquake.

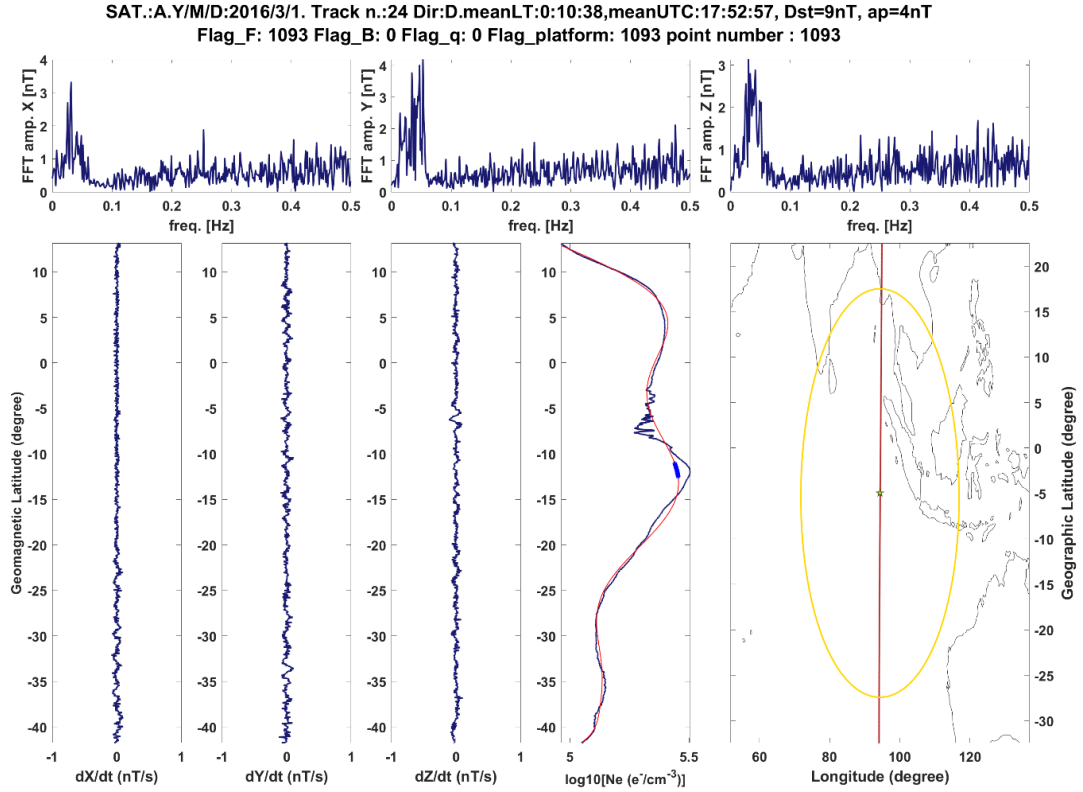




**Figure S13.** Swarm Alpha satellite nighttime track 23 of 2016-02-28 acquired 3 days before the M7.9 Sumatra 2016 earthquake.



**Figure S14.** Swarm Alpha satellite nighttime track 26 of 2016-02-29 acquired 2 days before the M7.9 Sumatra 2016 earthquake.

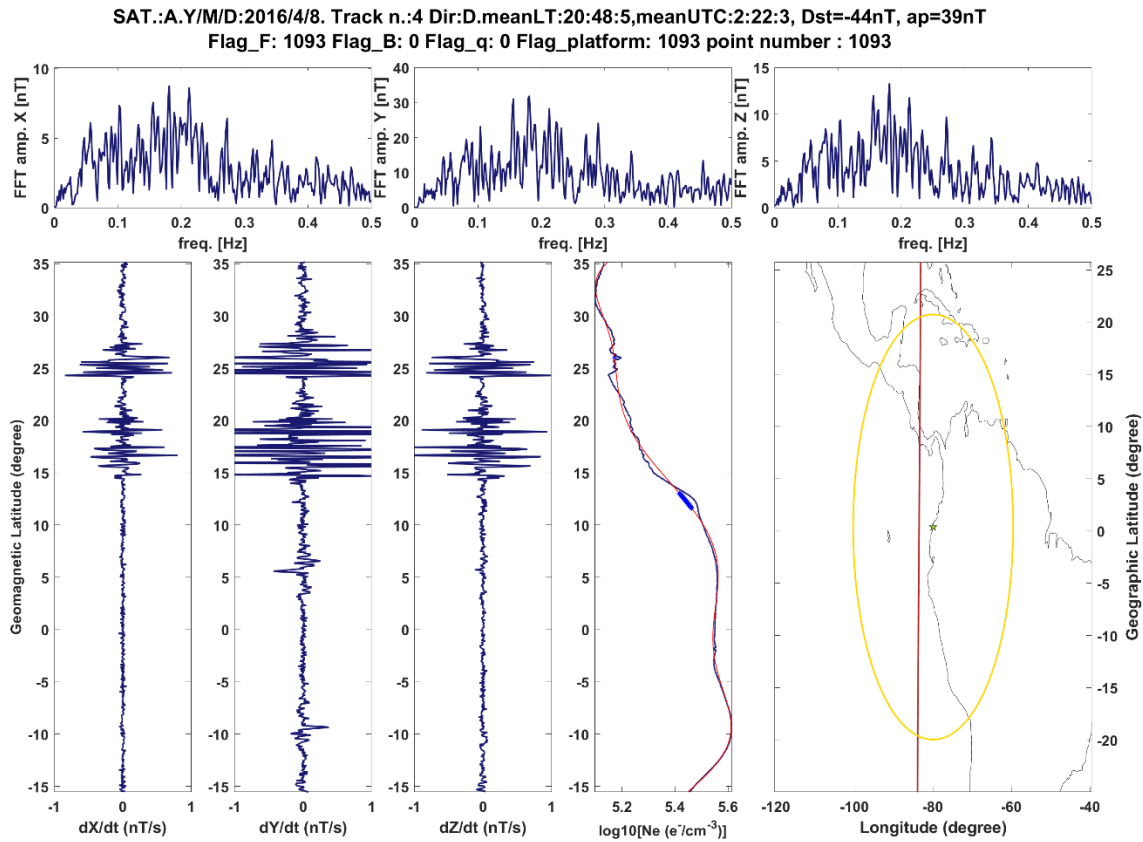


**Figure S15.** Swarm Alpha satellite nighttime track 24 of 2016-03-01 acquired the day before the M7.9 Sumatra 2016 earthquake.

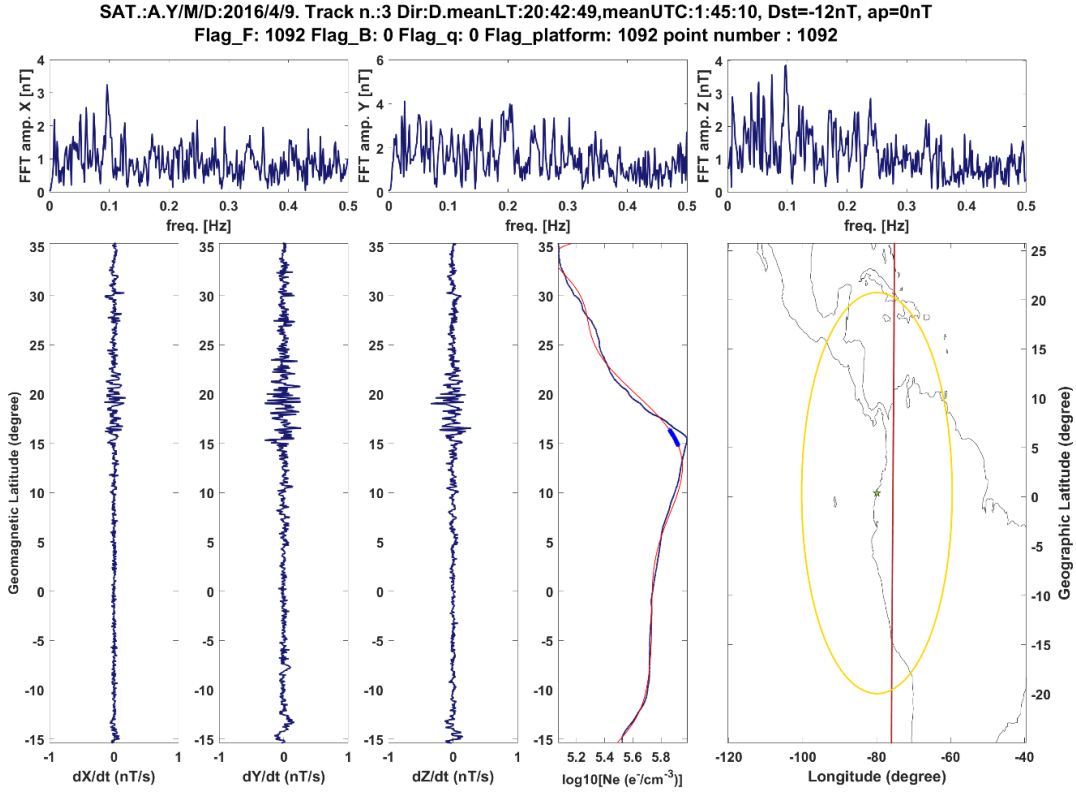
## Text S2. Swarm Alpha night-time tracks before M7.8 Ecuador 16-04-2016 earthquake.

In the following pictures (from Figure S16 to Figure S24) we show one Swarm Alpha night time track that crossed Dobrovosky's area for each day from 8 April 2016 until 16 April 2016, i.e. the day of occurrence of Mw=7.8 Ecuador 2016 earthquake at 23:58:36 UT.

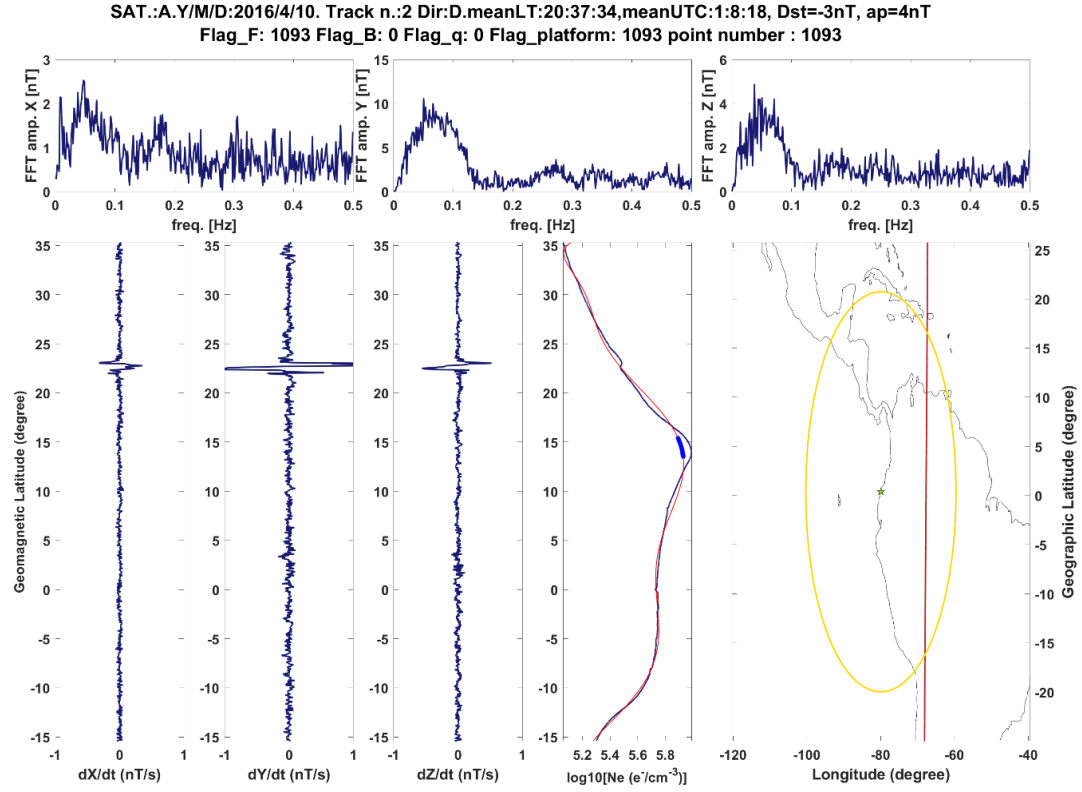
We noticed a clear external disturbance in Figure S21 with alteration of three components of magnetic field and contemporary depletion of electron density. A similar signal for magnetic field but curiously not associated with electron density disturbance is present in Figure S16. Both examples shows geomagnetic activity from the indices. The day of the earthquake, 22 hours before its occurrence Swarm Alpha satellite detected an anomaly (see Figure S24) in Y-East component of magnetic field, it is interesting because it has the potential characterist to be considered a possible precursor, even if little far from the imminent epicenter but inside the earthquake preparation area as defined by Dobrovosky.



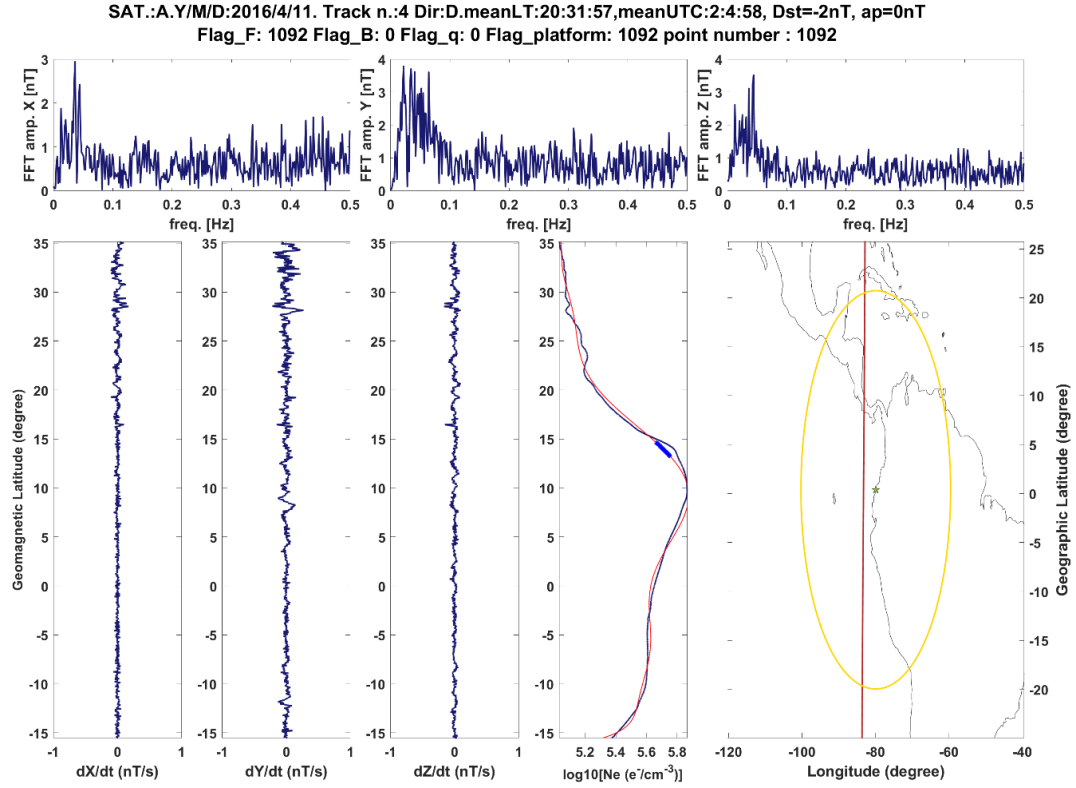
**Figure S16.** Swarm Alpha satellite nighttime track 4 of 2016-04-8 acquired 8 days before the M7.8 Ecuador 2016 earthquake.



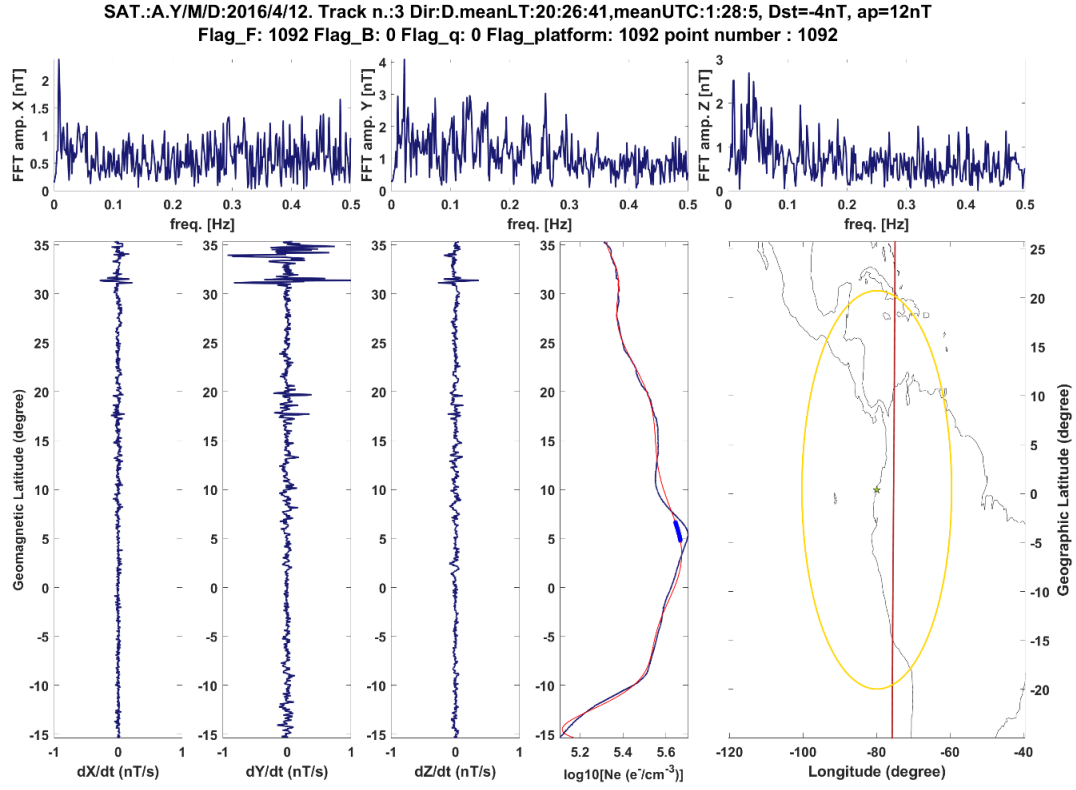
**Figure S17.** Swarm Alpha satellite nighttime track 3 of 2016-04-9 acquired 7 days before the M7.8 Ecuador 2016 earthquake.



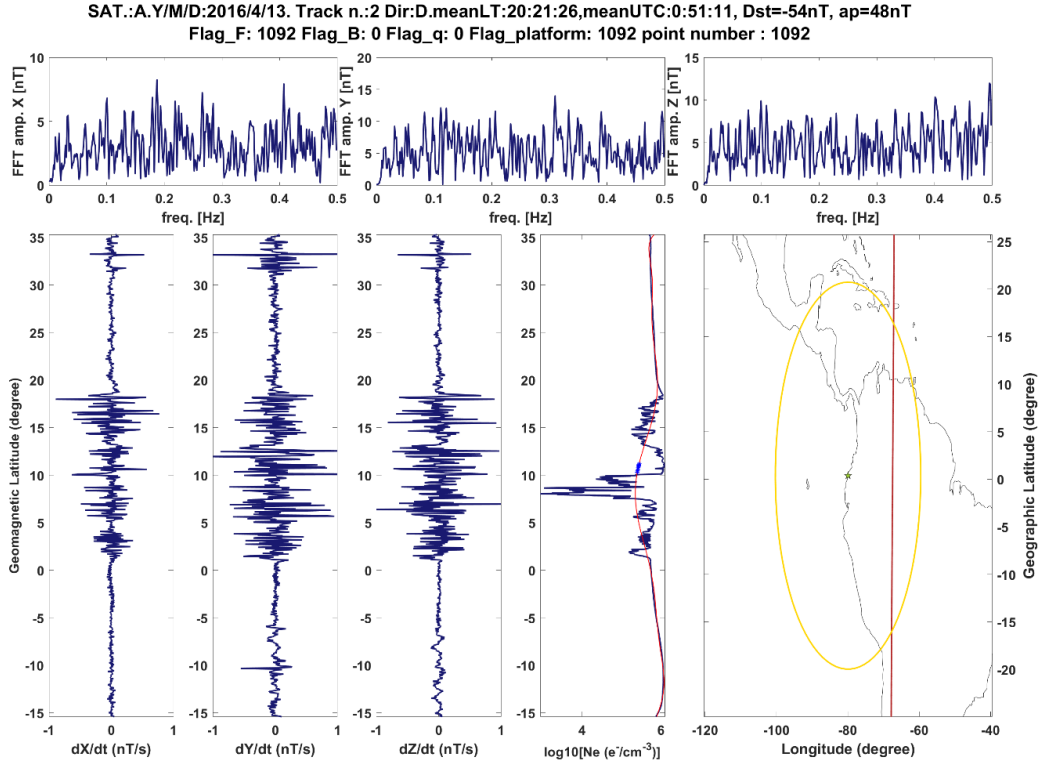
**Figure S18.** Swarm Alpha satellite nighttime track 2 of 2016-04-10 acquired 6 days before the M7.8 Ecuador 2016 earthquake.



**Figure S19.** Swarm Alpha satellite nighttime track 4 of 2016-04-11 acquired 5 days before the M7.8 Ecuador 2016 earthquake.

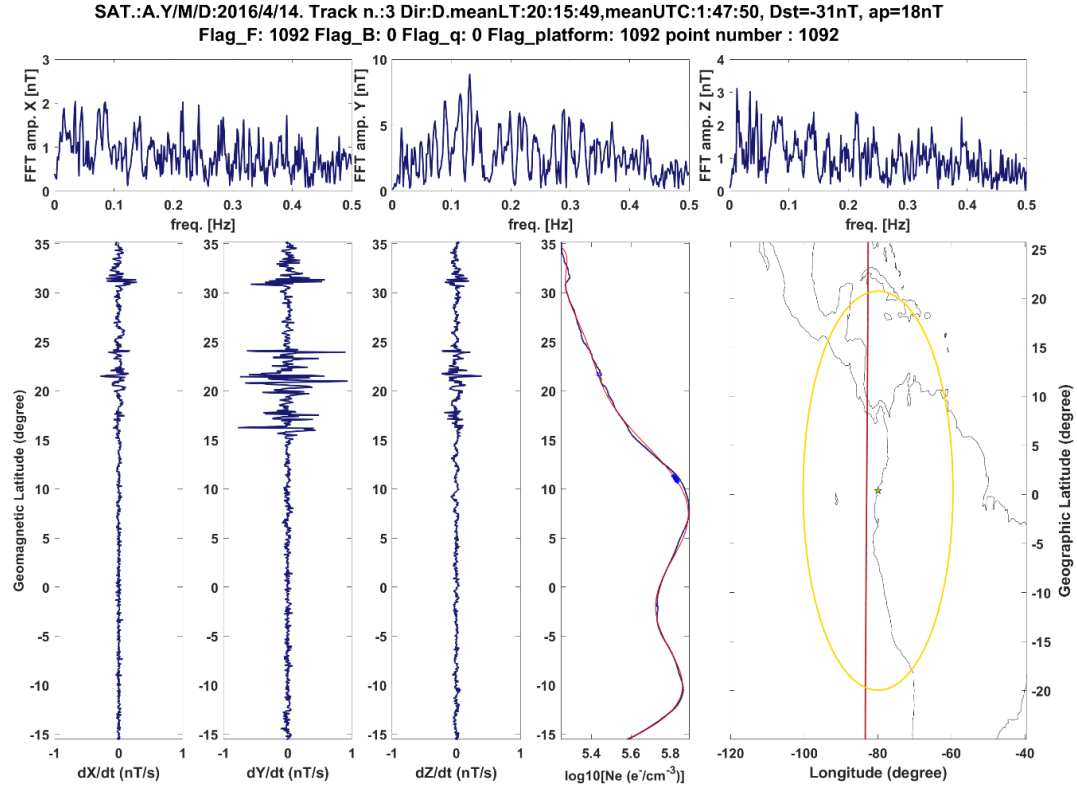


**Figure S20.** Swarm Alpha satellite nighttime track 3 of 2016-04-12 acquired 4 days before the M7.8 Ecuador 2016 earthquake.

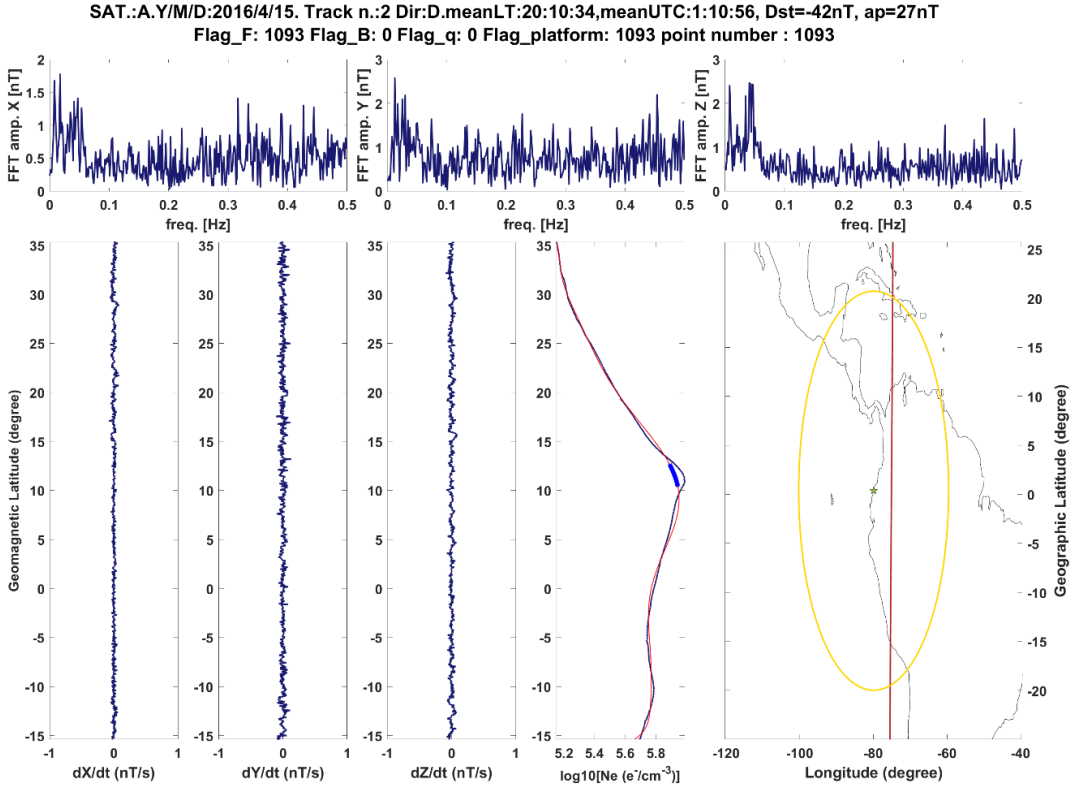


**Figure S21.** Swarm Alpha satellite nighttime track 5 of 2016–04–13 acquired 3 days before the M7.8 Ecuador 2016 earthquake. The geomagnetic condition at this time were disturbed (Dst = -54 nT and ap = 48 nT); the activity level was G1 – Minor storm.

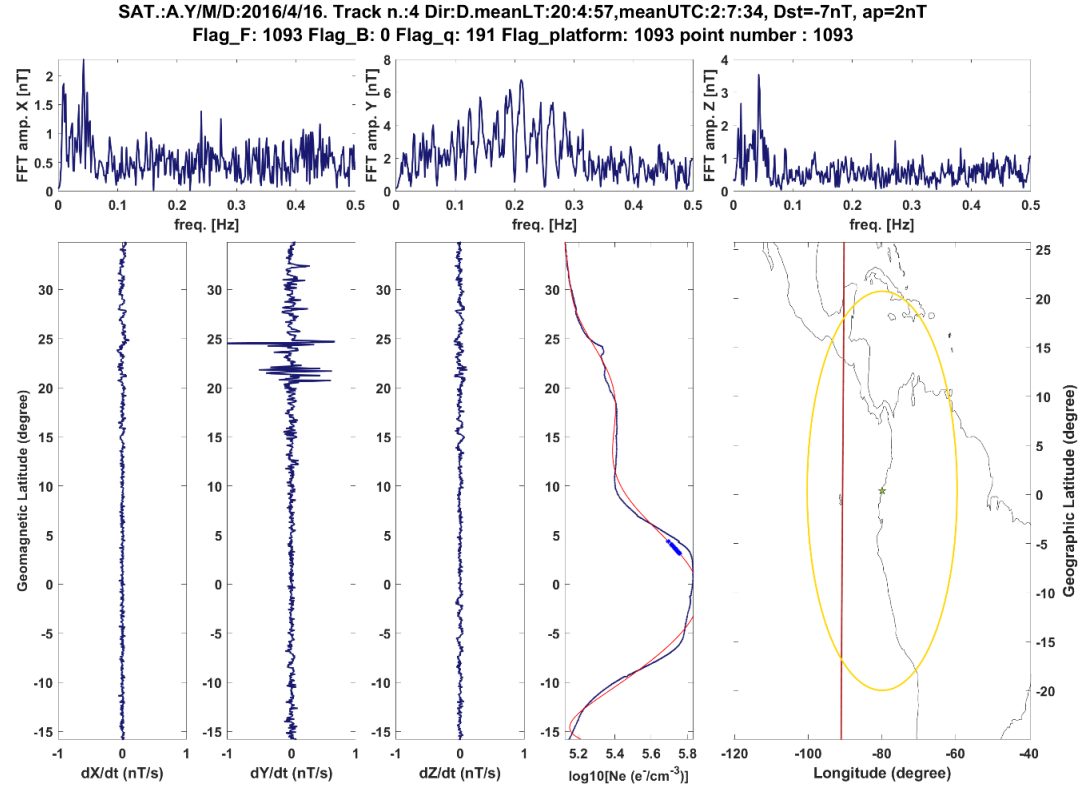




**Figure S22.** Swarm Alpha satellite nighttime track 3 of 2016-04-14 acquired 2 days before the M7.8 Ecuador 2016 earthquake.



**Figure S23.** Swarm Alpha satellite nighttime track 2 of 2016-04-15 acquired the day before the M7.8 Ecuador 2016 earthquake.



**Figure S24.** Swarm Alpha satellite nighttime track 4 of 2016-04-16 acquired 22 hours before the M7.8 Ecuador 2016 earthquake.

### Text S3. Performance metrics.

F1-score is a metric that combines precision and recall. It is usually described as the harmonic mean of both. Thus, the class imbalance is countered by weighting classes according to their sample proportion:

$$F_1 = \sum_i 2 * w_i \frac{PR_i * SN_i}{PR_i + SN_i} \quad (S1)$$

where  $i$  is the class index, and  $w_i = n_i / N$  is the proportion of samples of class  $i$ , with  $n_i$  being the number of samples of the  $i$ -th class and  $N$  being the total number of samples.

The Accuracy (ACC) is defined as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (S2)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives, respectively.

Beyond the metrics mentioned above, which emphasize positives, Matthews correlation coefficient (MCC) was also calculated [1]. MCC is calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (S3)$$

**Text S4. The SafeNet model structure.**

The SafeNet model structure mainly consists of preprocessing layer, convolutional layer, Bi-LSTM layer and softmax layer, the model structure is shown in Figure 4 of the main text.

Pre-processing layer: The input data are first normalized and normalized. The specific pre-processing methods are as follows.

(1) Normalize the numerical features with the formulas shown in Equations (S4), (S5) and (S6).

$$r'_{ij} = \frac{r_{ij} - AVG_j}{STAD_j} \quad (S4)$$

$$AVG_j = \frac{1}{n} (r_{1j} + r_{2j} + \dots + r_{nj}) \quad (S5)$$

$$STAD_j = \frac{1}{n} (|r_{1j} - AVG_j| + |r_{2j} - AVG_j| + \dots + |r_{nj} - AVG_j|) \quad (S6)$$

$r_{ij}$  is the numerical feature after processing, and the standardized value is  $r'_{ij}$ , where  $AVG_j$  is the mean value and  $STAD_j$  denotes the mean absolute deviation. If  $AVG_j$  is equal to 0, then  $r'_{ij} = 0$ ; if  $STAD_j$  is equal to 0, then  $r'_{ij} = 0$ .

(2) Normalization process: the normalized values are normalized to the interval [0, 1] and the normalized value of  $r'_{ij}$  is  $r''_{ij}$ . The formula is shown in (S7)

$$\begin{cases} r''_{ij} = \frac{r'_{ij} - r_{\min}}{r_{\max} - r_{\min}} \\ r_{\max} = \max \{ r'_{ij} \} \end{cases} \quad (S7)$$

Convolution layer: the convolution layer performs a convolution operation on  $V_s$  to produce a new feature  $h_i^d$  according to Equation (S8):

$$h_i^d = f(W_d \cdot V_i + b_d) \quad (S8)$$

Where,  $f$  is the ReLU function: for the specific features  $V_i$  of the record in record  $V_s$  for the convolution operation, in order to extract more comprehensive local features, set  $d$  different convolution kernel size  $W$  for feature extraction of  $V_i$  respectively,  $W_d$  indicates that the convolution kernel size is  $d$ ;  $b_d$  is the bias. After the specific convolution kernel size completes all the features in  $V_s$  are convolved, the output features are  $H^d$  equation as in Equation (S9):

$$H^d = [h_1^d, h_2^d, \dots, h_{n-d_r+1}^d] \quad (S9)$$

The features  $H^d$  obtained from the convolution operation of all convolution kernel sizes are superimposed to obtain the feature sequence  $H_s = [h_1, h_1, \dots, h_{n-d_r+1}]$ .

Pooling layer: pooling operation is performed on feature sequence  $H_s$ . The common pooling is divided into two types: average pooling and maximum pooling. In this paper, we adopt the average pooling method, and the specific process is shown in Equation (S10):

$$p^{dM} = chunkAve\{H^d\} = [h_{m_1}^d, h_{m_2}^d, \dots, h_M^d] \quad (S10)$$

Given a block number  $M$  in advance, divide  $H^d$  into  $M$  sub-blocks, and splice the average value in each sub-block, thus obtaining a feature vector  $p^{dm}$  of length  $M$ . The average value of all blocks is spliced to obtain  $P_s = [p_{m_1}, p_{m_2}, \dots, p_M]$ .  $p_{m_i}$  is the vector obtained after the average pooling of the block  $m_i$ .

BiLSTM layer: In order to capture long-range dependent features,  $P_s$  is fed into the BiLSTM model, which consists of LSTM modules connected in two directions with multiple shared weights. At each time step, the output of the BiLSTM module will be controlled by a forgetting gate ( $f_t$ ), an input gate ( $i_t$ ), an output gate ( $o_t$ ) and a cell state update together, each represented by the output  $b_{t-1}$  of the previous module and the input  $p_t$  at the current moment, and the three gates work together to complete the selection of attribute information, forgetting and cell state update. At time step  $t$ , feature extraction of  $p_t$  is performed using the forward part of the BiLSTM module with the formula as in Equation (S11):

$$\text{LSTM} \Rightarrow \begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, p_t] + b_i) \\ f_t = \sigma(W_f \cdot [h_{t-1}, p_t] + b_f) \\ q_t = \tanh(W_q \cdot [h_{t-1}, p_t] + b_q) \\ o_t = \sigma(W_o \cdot [h_{t-1}, p_t] + b_o) \\ c_t = f_t * c_{t-1} + i_t * q_t \\ b_t = o_t * \tanh(c_t) \end{cases} \quad (S11)$$

At time step  $t$ , feature extraction is performed using the inverse part of the BiLSTM module pair with the formula as in Equation (S12):

$$\text{LSTM} \Rightarrow \begin{cases} i_t = \sigma(W_i \cdot [h_{t+1}, p_t] + b_i) \\ f_t = \sigma(W_f \cdot [h_{t+1}, p_t] + b_f) \\ q_t = \tanh(W_q \cdot [h_{t+1}, p_t] + b_q) \\ o_t = \sigma(W_o \cdot [h_{t+1}, p_t] + b_o) \\ c_t = f_t * c_{t+1} + i_t * q_t \\ b_t = o_t * \tanh(c_t) \end{cases} \quad (\text{S12})$$

Where,  $\sigma$  is the sigmoid activation function;  $\tanh$  is the hyperbolic tangent function;  $*$  is the element multiplication operation,  $i_t$  is the selection operation of the input information to control the input process of the information,  $f_t$  is the forgetting operation of the information that needs to be forgotten in the previous module to control the forgetting process of the information.  $c_t$  is used to determine which information should be stored in the current cell state and to complete the control of the information storage.  $o_t$  is the output gate to select the output information and control the output information.

At time step  $t$ , the final output feature vector  $P_t$  of the BiLSTM layer is:

$$P_t = [\text{Forward LSTM}, \text{Backward LSTM}] \quad (\text{S13})$$

To obtain more accurate classification accuracy, the output of BiLSTM is fed to the fully connected layer. In the fully connected layer, the processing is publicized as Equations (S14), (S15) and (S16):

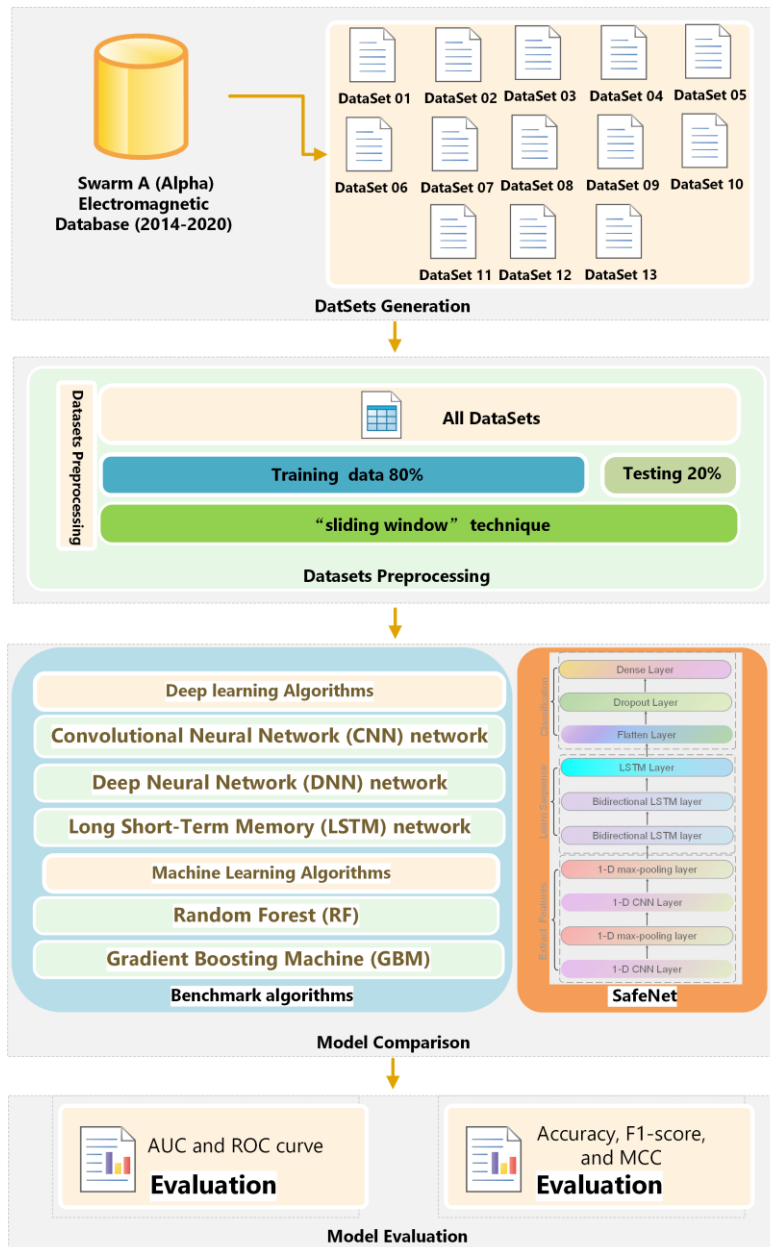
$$u_t = \tanh(W_w P_t + b_w) \quad (\text{S14})$$

$$a_t = \text{soft max}(u_t^T, u_w) \quad (\text{S15})$$

$$v = \sum a_t P_t \quad (\text{S16})$$

where  $u_t$  is the attribute representation of  $P_t$ ,  $u_w$  is the context vector,  $a_t$  is the importance weight, and  $v$  is the high-level representation obtained by weighted summation of importance over  $P_t$ .  $u_w$  is the high-level representation obtained by weighted summation of importance over

Finally, the output result  $v$  is fed into the softmax classifier to get the classification result.



**Figure S25.** The flowchart of the proposed deep learning framework.



**Table S1. Search space of parameters for the SafeNet model.**

Modules.	Parameters	Search Space
CNN	number of layers	hp.choice("num_cnn_layers", [2, 3, 4])
	number of filters	hp.choice("num_filters_cnn", [2, 3, 4])
	size of kernel	hp.choice("size_kernel_cnn", [2, 3])
	activation function	hp.choice("activation_cnn", ["relu", "tanh", "sigmoid"])
LSTM	number of layers	hp.choice("num_lstm_layers", [1, 2, 3, 4, 5])
	number of units for each layer	hp.choice("units_lstm_layers", [64, 128, 256, 512])
Fully Connected (FC)	number of layers	hp.choice("num_fc_layers", [2, 3, 4, 5, 6])
	number of units for each layer	hp.choice("units_fc_layers", [64, 128, 256, 512])
	activation function	hp.choice("activation_fc", ["relu", "tanh", "sigmoid"])
Common	batch_size	hp.choice("batch_size", [10, 20, 40, 60, 80, 100])
	number of epochs	hp.choice("nb_epochs", [10, 50, 100])
	dropout	hp.choice("dropout", [0.25, 0.5, 0.75])
	scale	hp.choice('scale', [0, 1])
	normalize	hp.choice('normalize', [0, 1])

For more details on the search spaces in the table, please refer to the Hyperopt document available at ([http://hyperopt.github.io/hyperopt/getting-started/search\\_spaces/](http://hyperopt.github.io/hyperopt/getting-started/search_spaces/)).

hp.choice (label, options) returns one of the options, which should be a list or tuple.

## Reference

- [1] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim Biophys Acta*, vol. 405, no. 2, pp. 442-51, Oct 20 1975, doi: 10.1016/0005-2795(75)90109-9.