Machine Learning for Solar Flare Detection

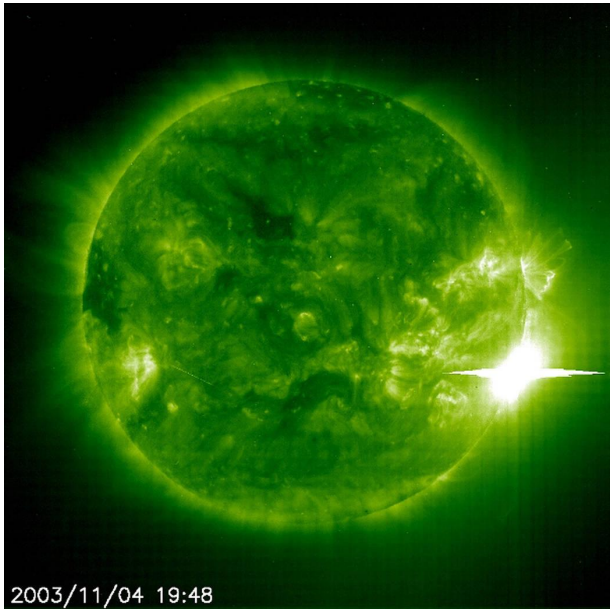
Raymond Blaha, Paul Hwang, David Nezelek
M.S. Students in Applied Data Science
New College of Florida
December 19, 2023

Introduction

Presenters: Raymond Blaha, Paul Hwang, and David Nezelek, second-year Master's students in Prof. Klingenberg's Practical Data Science course

Research task: to detect a solar flare, locate it within a known active region on the surface of the sun, and measure its properties.

Goal of the presentation: to present our findings from using machine learning to detect solar flares.



https://www.nasa.gov/image-article/what-solar-flare/

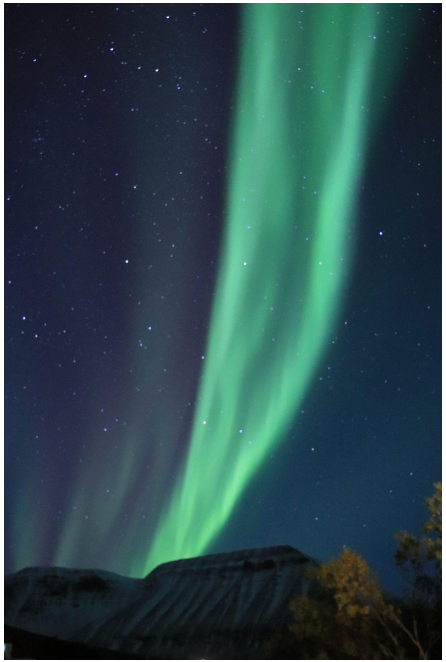
Why solar flares?

A solar flare is *an intense burst of radiation coming from the release of magnetic energy*.

Solar flares cause the awe-inspiring *aurora borealis*, visible at night from the Earth's upper latitudes.

Flares are associated with coronal mass ejections (CMEs): *large expulsions of plasma and magnetic field*.

Large CMEs can interfere with radio and satellite communication, and even electric power transmission.



Aurora borealis. Photo by David Nezelek

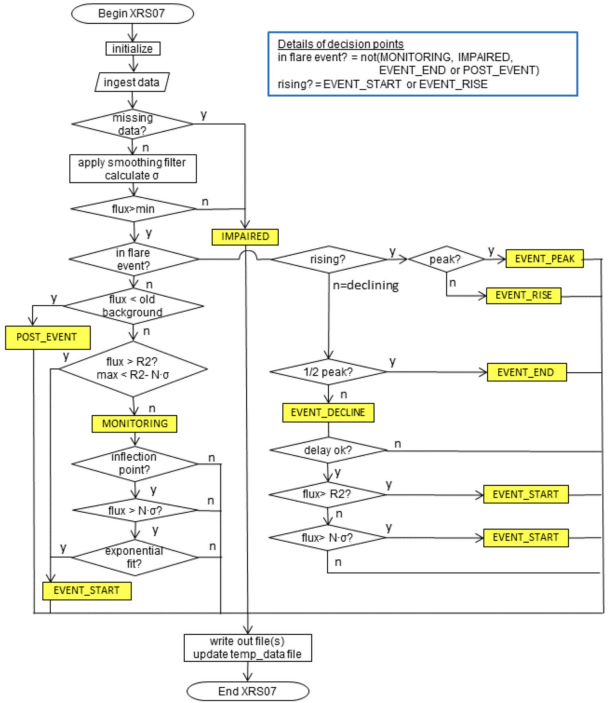
How are solar flares detected currently?

NASA uses a computational algorithm to detect and record solar flares.

Flares are divided into five classes by intensity: A, B, C, M, and the most powerful X flares.

The algorithm reliably finds the more powerful flares, but fails to detect many in classes A and B.

Recent advances in machine learning hold promise for improving upon this algorithm.



What data are used to detect flares?

EXIS:
Extreme Ultraviolet and X-ray Sensors

XRS

XRS 1-minute Averages

XRS 1-second Fluxes

XRS Daily Background

XRS Flare Location

XRS Flare Summary

The NASA algorithm uses satellite measurements of x-ray flux, the rate at which energy in the x-ray spectrum is emitted by the sun.

NASA provides these x-ray sensor (XRS) measurements to the public, as well as the time, location, and class of the flares identified by the algorithm.

Research Objective

To use machine learning to detect solar flares from NASA's XRS data.

A machine learning model which detects flares more reliably than the long-used computational algorithm would provide a more accurate database of flares, which would benefit all who research the causes of solar flares and the effects we feel on Earth.

Time Series Analysis

- Use Time Series to Classify Flares
  - Long-Short Term Memory (LSTM)
  - One-Dimensional Convolutional Neural Network (1D CNN)
  - Mixed Model

Time Series Preprocessing

- Scale Data from 0 to 1 using MinMax Scaler
- Format data to Time Series Format
  - Convert a table into a 3-D Data Frame

	Xrsa_flux	Xrsb_flux	Flare
Day1	1	2	0
Day2	3	4	1
Day3	5	6	0
Day4	7	8	1

	Xrsa_flux	Xrsb_flux	Flare
Day2	(1,3,5)	(2,4,6)	1
Day3	(3,5,7)	(4,6,8)	0

Time Series Methodology

```
model5 = tf.keras.models.Sequential([
  tf.keras.layers.Conv1D(filters=64, kernel_size=5, strides=1, padding="causal",
                         activation="relu", input_shape=(X_train.shape[1], X_train.shape[2])),
  tf.keras.layers.LSTM(64, return_sequences=True),
  tf.keras.layers.LSTM(64, return_sequences=False),
  tf.keras.layers.Dense(30, activation="relu"),
  tf.keras.layers.Dense(10, activation="relu"),
  tf.keras.layers.Dense(1)
])

optimizer = tf.keras.optimizers.SGD(learning_rate=1e-8, momentum=0.9)
model5.compile(loss=tf.keras.losses.Huber(),
               optimizer=optimizer, metrics=['accuracy'])
```

```
## LSTM
model = Sequential()
model.add(LSTM(64, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(LSTM(64, return_sequences=False, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dense(100, activation = 'relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

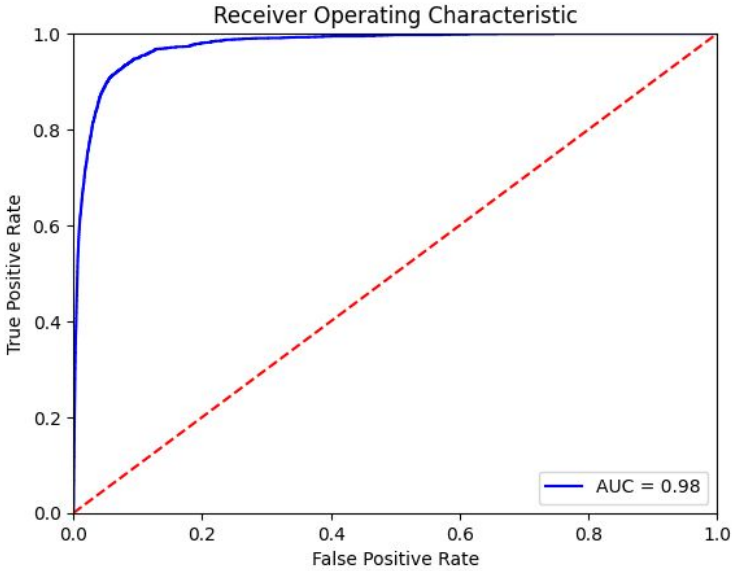
```
## 1D convolution
model3 = Sequential()
model3.add(Conv1D(64, 2, activation="relu", input_shape=(X_train.shape[1], X_train.shape[2])))
model3.add(Conv1D(64, 2, activation="relu", input_shape=(X_train.shape[1], X_train.shape[2])))
model3.add(Flatten())
model3.add(Dense(100, activation="relu"))
model3.add(Dense(100, activation="relu"))
model3.add(Dense(1, activation='sigmoid'))
model3.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

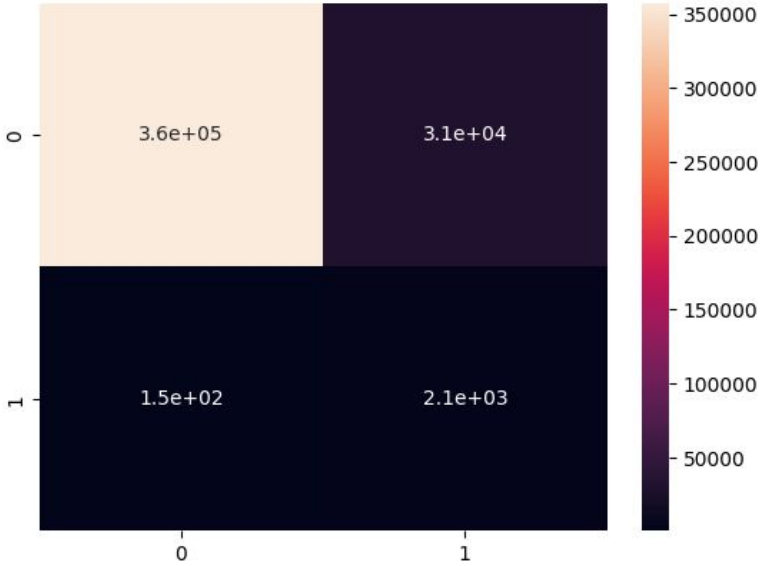
✓ 0.0s

Time Series Results

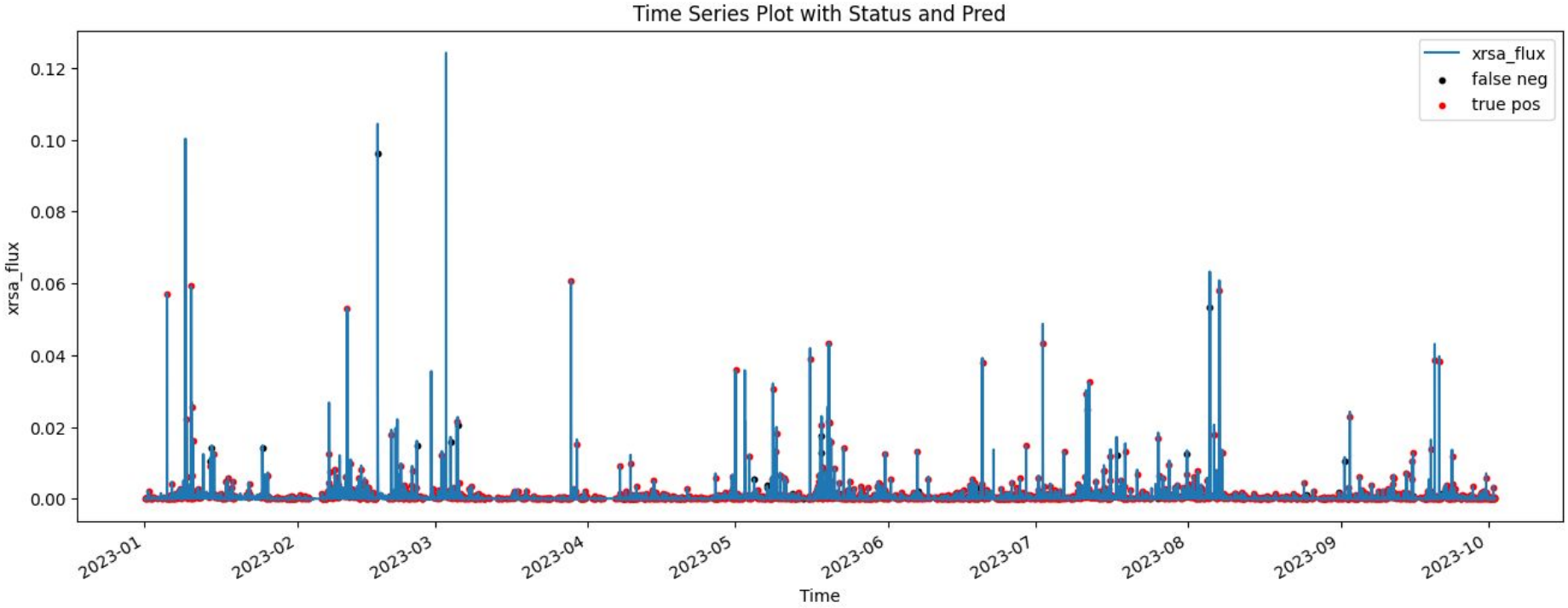
Classification Report:

	precision	recall	f1-score	support
0.0	1.00	0.92	0.96	388433
1.0	0.07	0.93	0.12	2300
accuracy			0.92	390733
macro avg	0.53	0.93	0.54	390733
weighted avg	0.99	0.92	0.95	390733

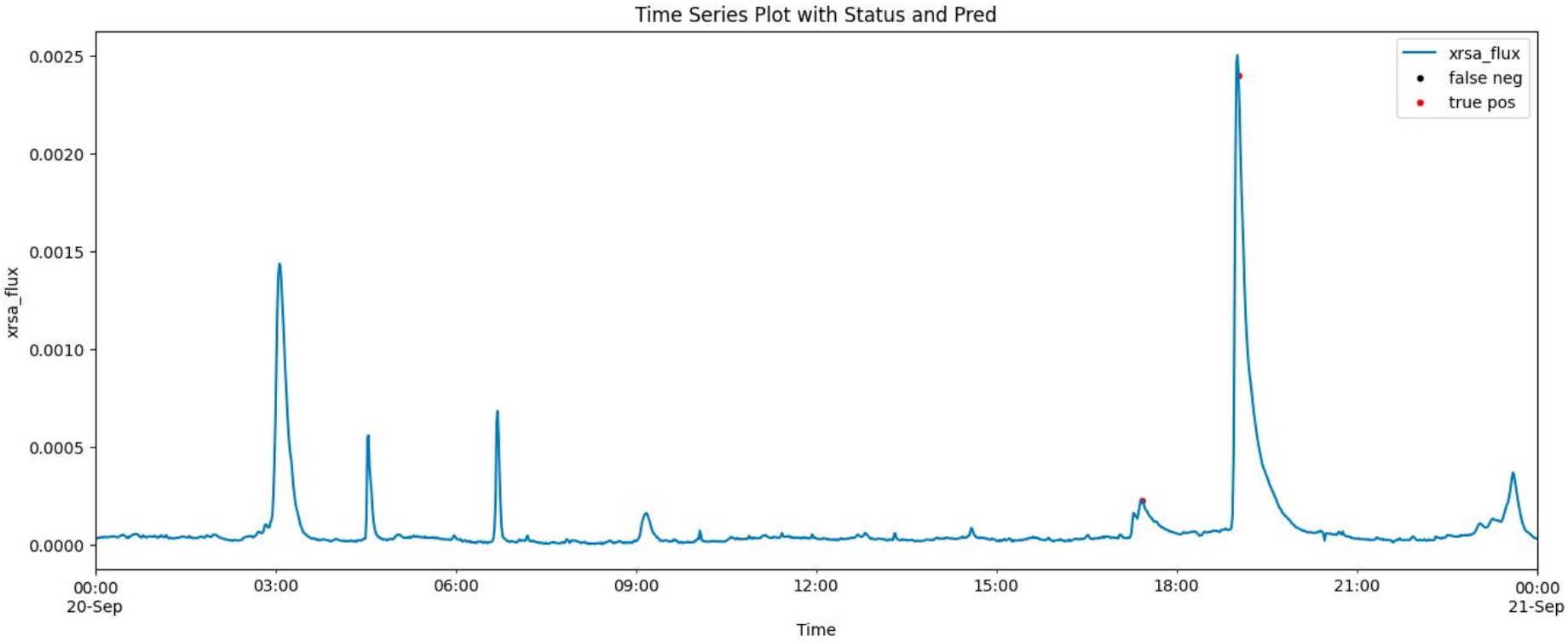




Time Series Results



Time Series Results



AutoEncoder Setup

- Purpose:
  - To detect anomalies in the temporal data.
  - Utilizing the autoencoder architecture for “feature extraction” and noise reduction.
- Sequence Generation:
  - Created 24-hour window of data for training and validation.

AutoEncoder Structure

- Encoder Structure
  - Input: Sequence length and feature count.
  - Layers: Conv1D with ReLu activation, MaxPooling1D, Dropout, and L2 Regularization.
- Latent Dimension
  - Latent Dimension (10).
- Decoder Structure
  - Input: Latent Dimension
  - Layers: Conv1D, Reshape, Batch Normalization, Conv1DTranspose.
  - Output: Conv1D layer to reconstruct the input sequence.

AutoEncoder Structure

```
# Model Architecture
latent_dim = 10

# Encoder
encoder_input = layers.Input(shape=(SEQUENCE_LENGTH, len(norm_cols)))
x = layers.Conv1D(16, 3, activation='relu', padding='same', kernel_regularizer=regularizers.l2(0.001))(encoder_input)
x = layers.MaxPooling1D(2)(x)
x = layers.Dropout(0.3)(x)
x = layers.Conv1D(32, 3, activation='relu', padding='same', kernel_regularizer=regularizers.l2(0.001))(x)
x = layers.MaxPooling1D(2)(x)
x = layers.Flatten()(x)
encoder_output = layers.Dense(latent_dim, activation='relu')(x)
encoder = Model(inputs=encoder_input, outputs=encoder_output)

# Decoder
decoder_input = layers.Input(shape=(latent_dim,))
x = layers.Dense((SEQUENCE_LENGTH//4)*32, activation='relu')(decoder_input)
x = layers.Reshape((SEQUENCE_LENGTH//4, 32))(x)
x = layers.BatchNormalization()(x)
x = layers.Conv1DTranspose(32, 3, strides=2, activation='relu', padding='same')(x)
x = layers.Conv1DTranspose(16, 3, strides=2, activation='relu', padding='same')(x)
decoder_output = layers.Conv1D(len(norm_cols), 3, activation='linear', padding='same')(x)
decoder = Model(inputs=decoder_input, outputs=decoder_output)
```

Anomaly Detection

- Detection Method
  - Using the reconstruction error that was generated from the Decoder to identify anomalies.
  - Threshold set on error distribution (95th percentile)

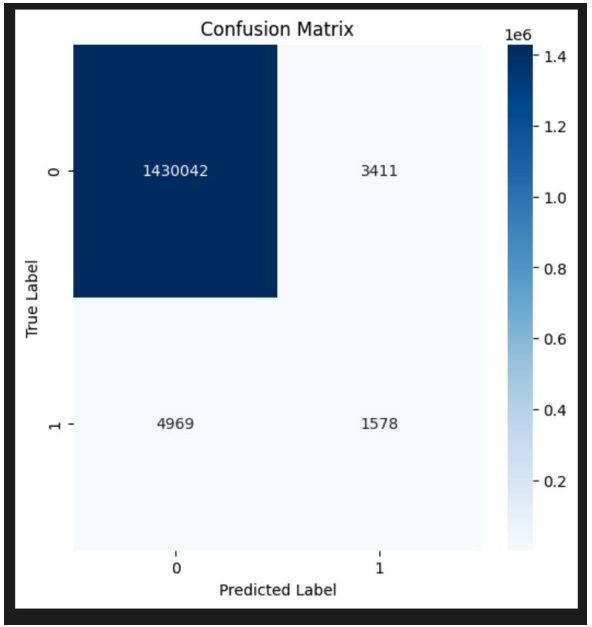
AutoEncoder Results

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1433453
1	0.32	0.24	0.27	6547
accuracy			0.99	1440000
macro avg	0.66	0.62	0.64	1440000
weighted avg	0.99	0.99	0.99	1440000

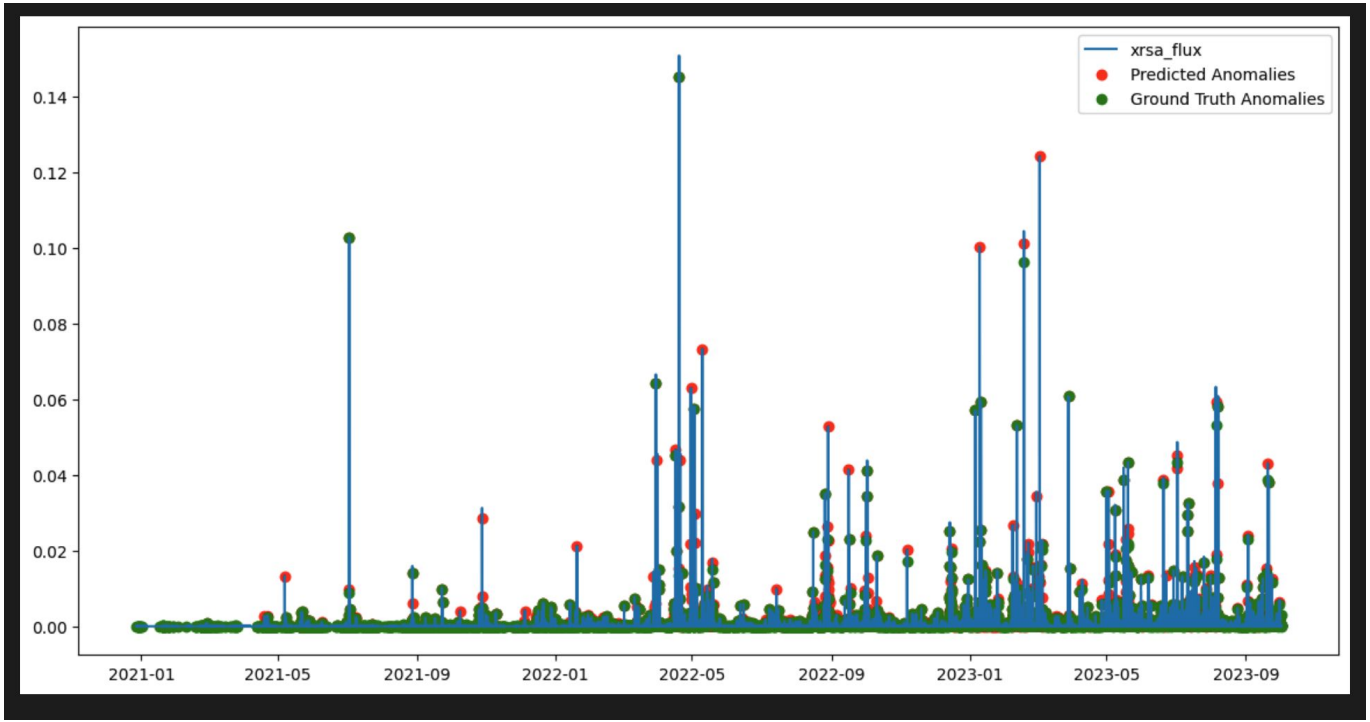
- **True Negatives:** Predicted the non-anomalous class correctly 1,430,042 times.
- **False Positives:** The model incorrectly predicted the anomalous class 3,411 times when it actually was a non-anomalous class.
- **False Negatives:** The model incorrectly predicted the non-anomalous class 4,969 times when it was actually the anomalous class.
- **True Positive:** The model correctly predicted the anomalous class 1,578 times.

Takeaways:

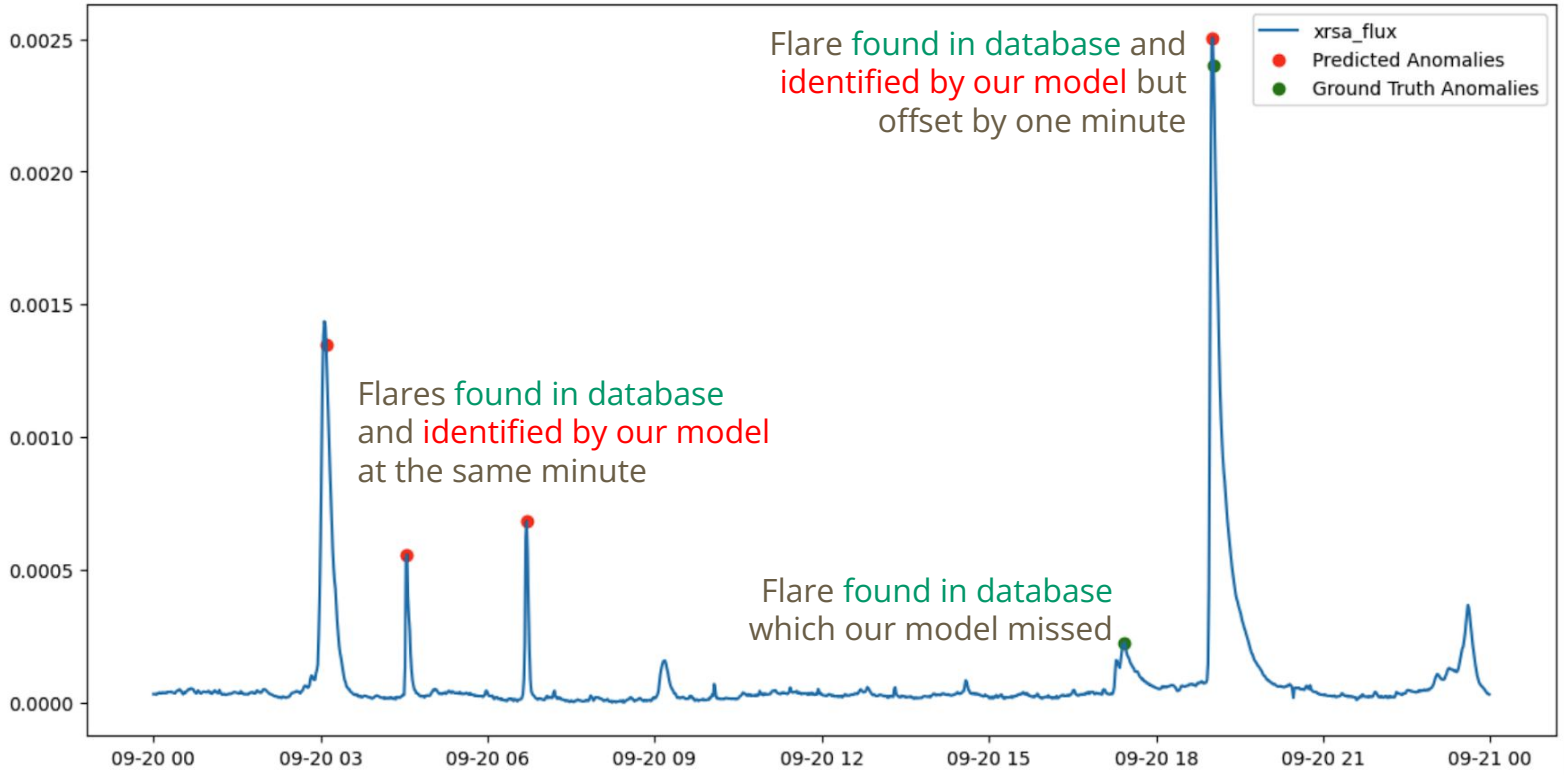
- Great at identifying majority class.
- Under performs on minority class.



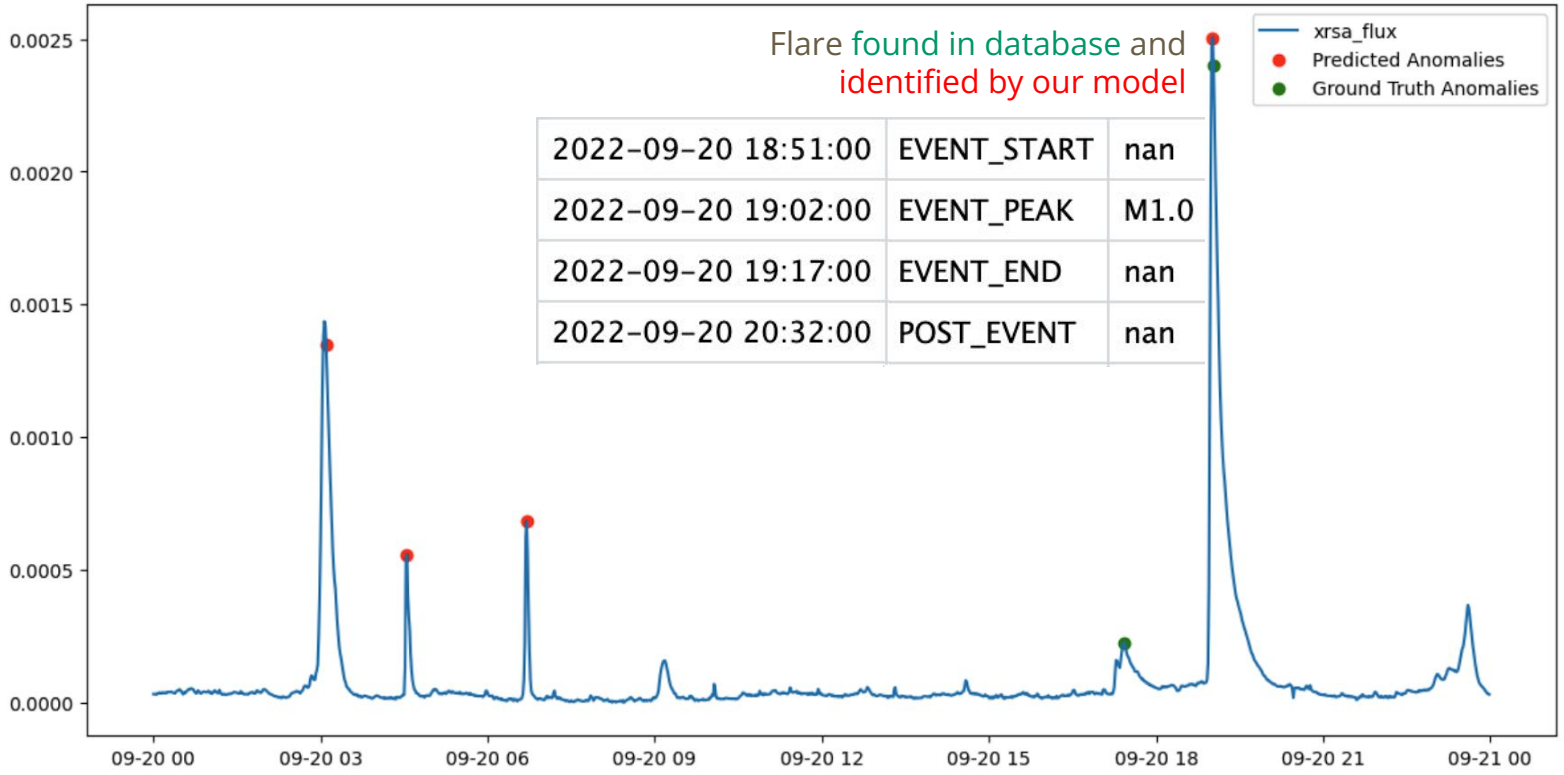
Comparing with Ground Truth on September 20, 2022



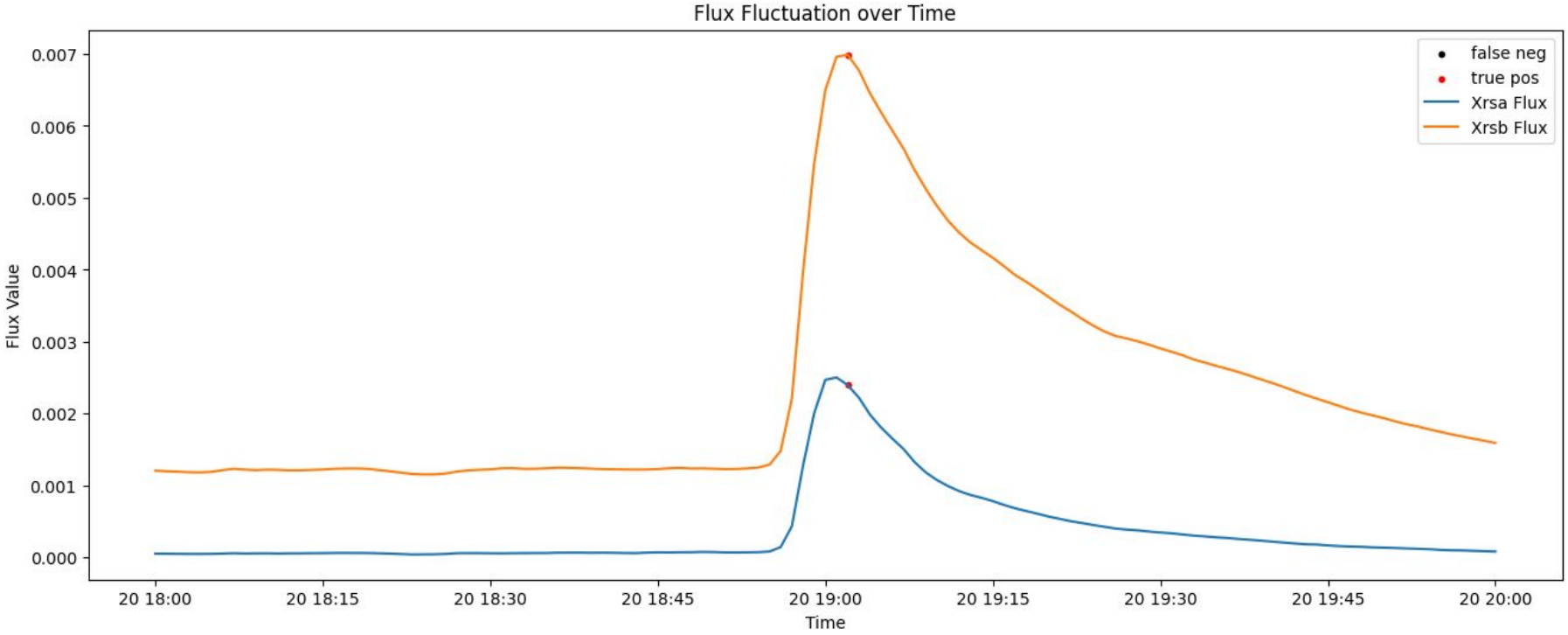
Flares and potential flares on 9/20/22



Examining a flare



Examining a flare



An error in date conversion

www.SolarMonitor.org

20 September 2022

Today's/Yesterday's NOAA Active Regions

NOAA Number	Latest Position	Hale Class	McIntosh Class	Sunspot Area [millionths]	Number of Spots	Recent Flares
13100	S22W87 (884",-360")	β/β	Cso/Dso	0070/0150	03/06	-
13102	S26W27 (391",-511")	β/β	Eki/Eki	0310/0320	24/25	C2.0(20:59) C4.5(18:33) C3.8(16:24) C7.7(14:51) M1.0(11:13) C2.4(09:18) C5.5(05:35) C2.2(02:52) / C3.3(20:22) C1.5(19:30) C2.5(13:46) C1.8(10:29)
13104	S11W80 (923",-201")	α/-	Axx/---	0010/----	01/--	-
13105	S18E53 (-727",-361")	β/-	Dao/---	0050/----	06/--	-
13103	S16W64 (826",-311")	/	/	/	/	-

Events not associated with currently named NOAA regions: C1.9(01:26) C1.9(02:14) C2.2(04:08) C4.8(06:05) C1.7(11:43) C1.6(16:51) C1.5(19:02) C2.5(19:59)

2022-09-20 18:51:00	EVENT_START	nan
2022-09-20 19:02:00	EVENT_PEAK	M1.0
2022-09-20 19:17:00	EVENT_END	nan
2022-09-20 20:32:00	POST_EVENT	nan

We could not find this flare in the records available online.

How to account for this discrepancy?

Returning to the source

We went back and checked the data file we used for analysis.

Our date conversion was off by 12 hours.

2022-09-20 18:51:00	EVENT_START	nan
2022-09-20 19:02:00	EVENT_PEAK	M1.0
2022-09-20 19:17:00	EVENT_END	nan
2022-09-20 20:32:00	POST_EVENT	nan

2022-09-20 18:51:00	1.758357e-06	EVENT_START	nan	202209210651
2022-09-20 19:02:00	1.013076e-05	EVENT_PEAK	M1.0	202209210651
2022-09-20 19:17:00	5.665353e-06	EVENT_END	nan	202209210651

Fortunately, this error is easy to correct.

Correcting the error

www.SolarMonitor.org

21 September 2022

Today's/Yesterday's NOAA Active Regions

NOAA Number	Latest Position	Hale Class	McIntosh Class	Sunspot Area [millionths]	Number of Spots	Recent Flares
13100	S24W91 (872",-383")	α/β	Hax/Cso	0080/0070	01/03	-
13102	S26W40 (553",-497")	β/β	Eko/Eki	0280/0310	16/24	-
13105	S14E48 (-690",-306")	β/β	Dhi/Dao	0350/0050	11/06	-
13106	S10E42 (-631",-251")	β/-	Bxo/---	0010/-----	02/--	-
13103	S16W78 (898",-284")	/	/	/	/	-
13104	S11W91 (937",-178")	/α	/Axx	/0010	/01	-

Events not associated with currently named NOAA regions: C2.8(05:12) C3.4(10:37) C1.4(12:42) C1.7(18:14) M1.0(06:51) C1.9(01:26) C1.9(02:14) C2.2(02:52) C2.2(04:08) C5.5(05:35) C2.4(09:18) C7.7(14:51) C3.8(16:24) C4.5(18:33) C2.0(20:59) M1.0(11:13)

2022-09-20 18:51:00	EVENT_START	nan
2022-09-20 19:02:00	EVENT_PEAK	M1.0
2022-09-20 19:17:00	EVENT_END	nan
2022-09-20 20:32:00	POST_EVENT	nan

Examining a flare

Events not associated with currently named NOAA regions: C2.8(05:12) C3.4(10:37) C1.4(12:42) C1.7(18:14) M1.0(06:51)

If our flare is not associated with a named active region, where is it?

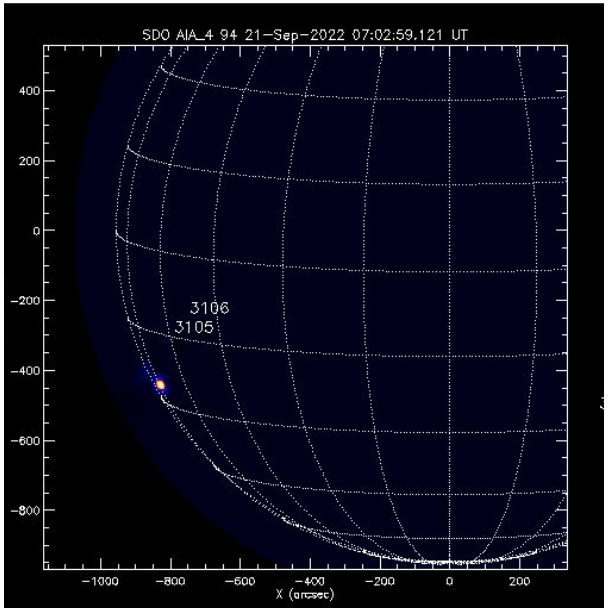
-847.3016
-422.4999

coordinates from
GOES flare location file

In an emerging active region yet to be named.

Event#	EName	Start	Stop	Peak	GOES Class	Derived Position
2	gev_20220921_0651	2022/09/21 06:51:00	07:18:43	07:02:00	M1.0	S25E73 (3107)

table and image from lmsal.com
Lockheed Martin Solar & Astrophysics Laboratory



Examining a flare

The region was named the following day.

History of Active Region 13107

Description of the Active Region and a summary of flaring activity:

DATE	TYP	AREA	NSPOT	ZMCINT	MCLASS	C	M	X	TOTAL	SMON
2022-09-22	A	120	2	Dso	β	0	0	0	0	Link
2022-09-23	A	190	6	Cao	β	0	0	0	0	Link
2022-09-24	A	230	10	Cao	β	0	0	0	0	Link
2022-09-25	A	240	17	Fai	βγ	0	0	0	0	Link
2022-09-26	A	240	17	Eai	β	0	0	0	0	Link
2022-09-27	A	220	33	Esi	β	0	0	0	0	Link
2022-09-28	A	200	27	Eai	βγ	0	0	0	0	Link
2022-09-29	A	190	17	Eai	βγ	0	0	0	0	Link
2022-09-30	A	160	8	Eai	β	0	0	0	0	Link
2022-10-01	A	100	7	Cai	β	0	0	0	0	Link
2022-10-02	A	30	6	Cri	β	0	0	0	0	Link
2022-10-03	A	30	2	Hsx	α	0	0	0	0	Link

table from the Heliophysics Event Catalog, **helio.mssl.ucl.ac.uk**

Examining a flare

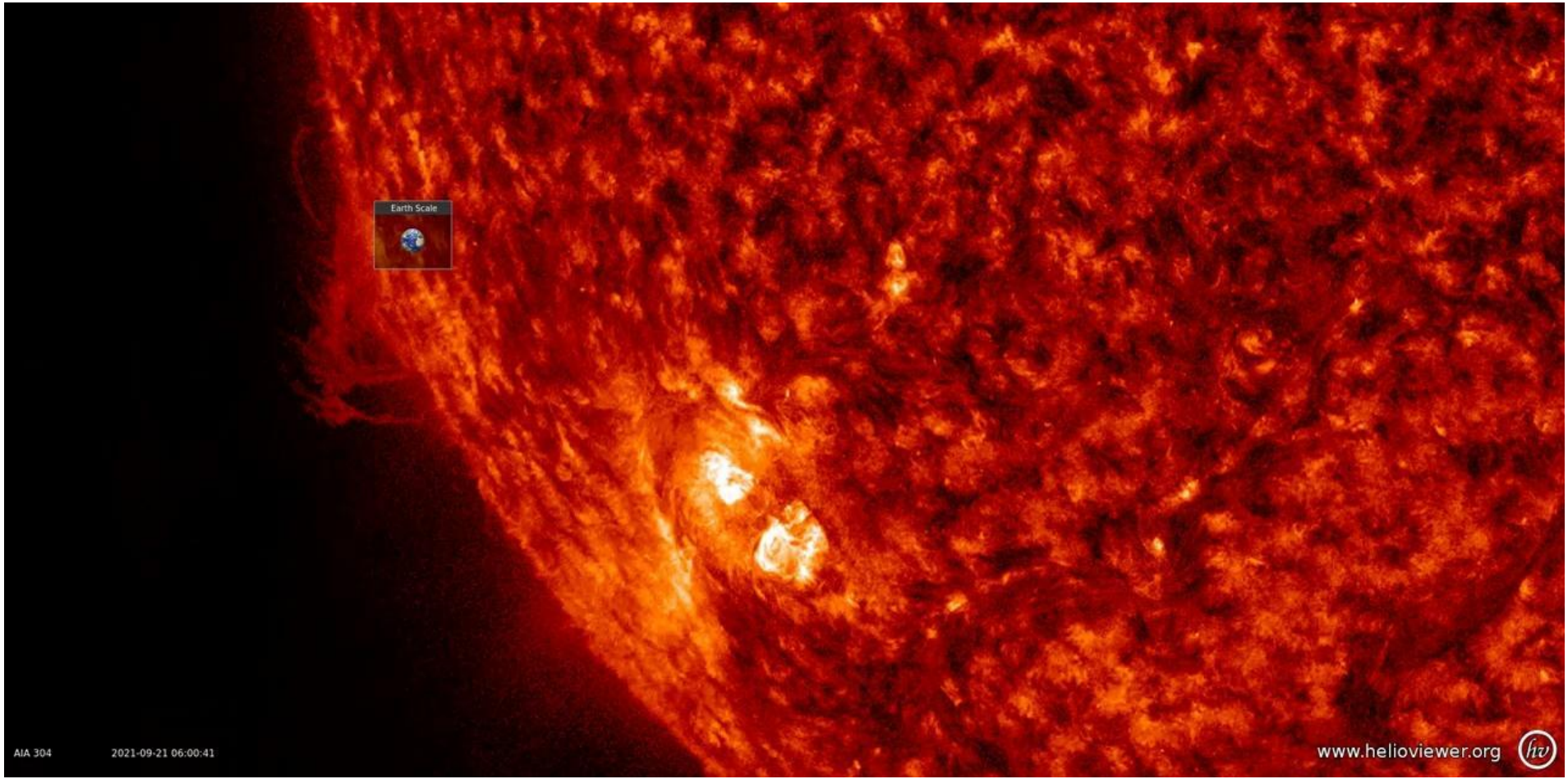
Energy released per square meter estimated by Riemann sum of flux values between the start and end times.

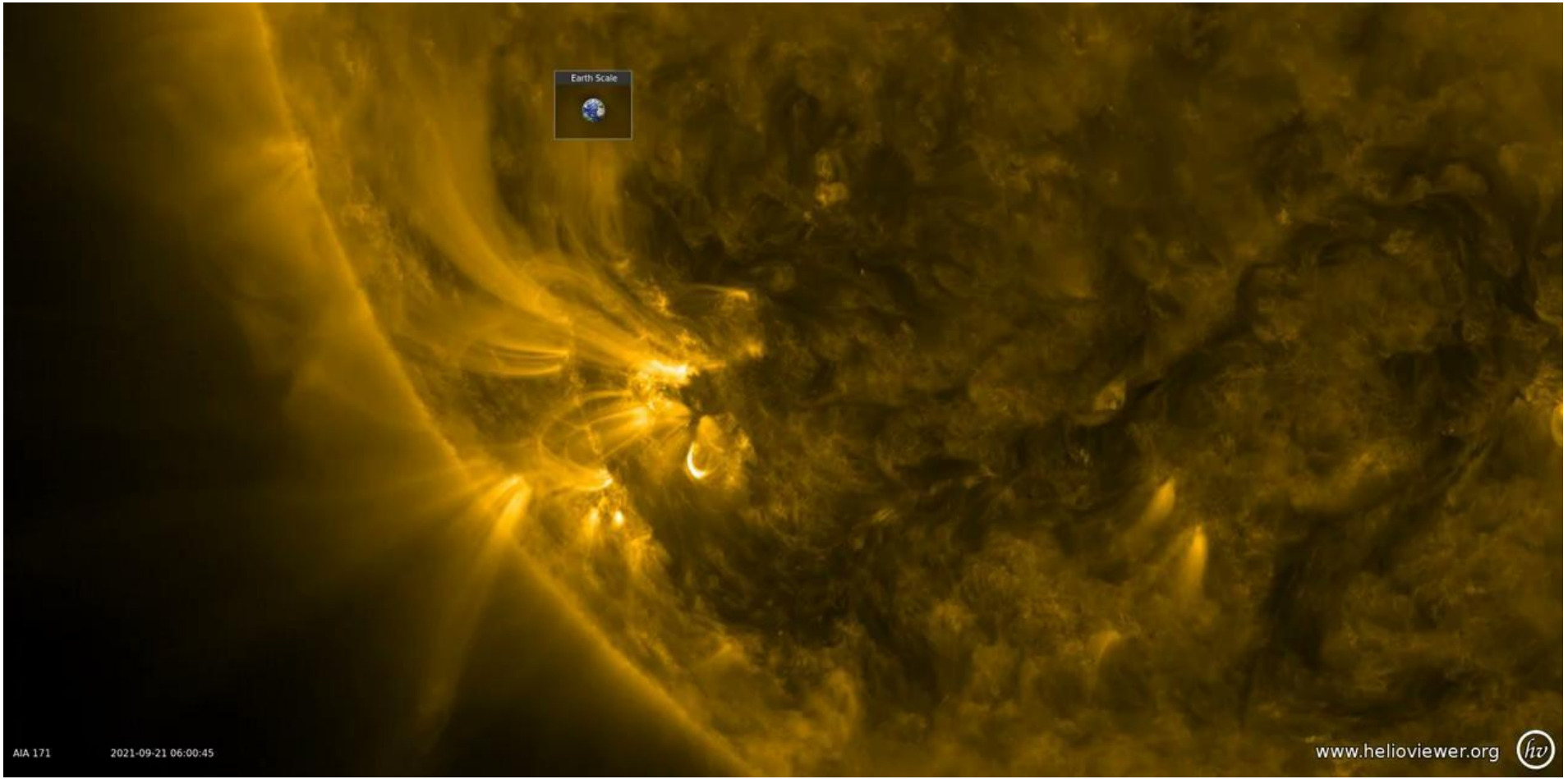
Sum of minute by minute flux values
= 1.8×10^{-05} W/m^2
x 60 seconds/minute = 1.1×10^{-03} Joules/m^2

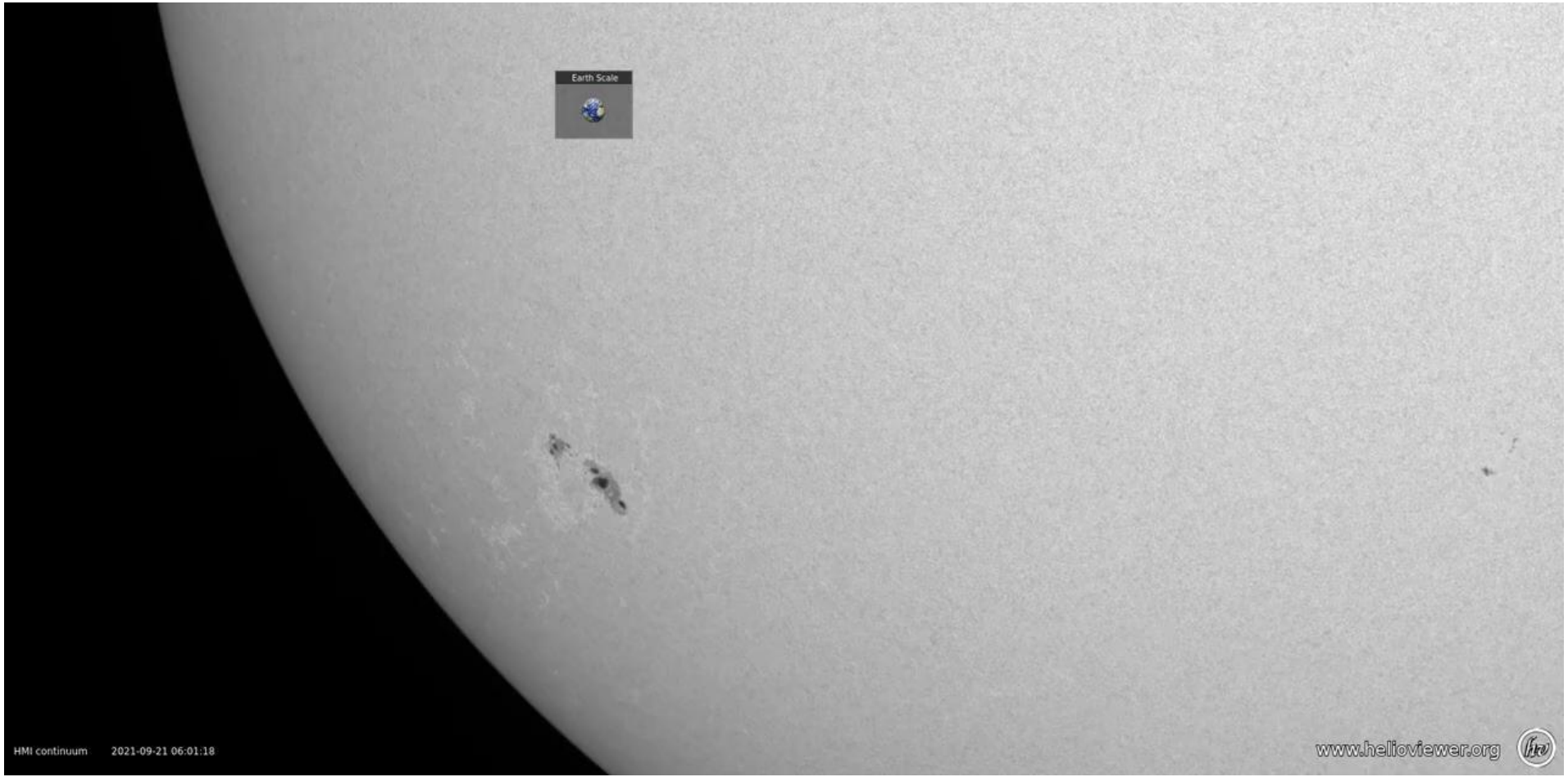
Estimated baseline energy released per square meter over this time
4.5×10^{-7} x 26 minutes x 60 seconds = 7.0×10^{-4} J/m^2

Estimated energy released by flare per square meter
= difference = 4×10^{-3} J/m^2

	time	xrsa_flux
2955291	2022-09-20 18:50:00	4.613691e-08
2955292	2022-09-20 18:51:00	4.367525e-08
2955293	2022-09-20 18:52:00	4.341389e-08
2955294	2022-09-20 18:53:00	4.300169e-08
2955295	2022-09-20 18:54:00	4.464934e-08
2955296	2022-09-20 18:55:00	5.014473e-08
2955297	2022-09-20 18:56:00	8.662342e-08
2955298	2022-09-20 18:57:00	2.602304e-07
2955299	2022-09-20 18:58:00	7.649394e-07
2955300	2022-09-20 18:59:00	1.208287e-06
2955301	2022-09-20 19:00:00	1.486952e-06
2955302	2022-09-20 19:01:00	1.505989e-06
2955303	2022-09-20 19:02:00	1.443556e-06
2955304	2022-09-20 19:03:00	1.336322e-06
2955305	2022-09-20 19:04:00	1.193963e-06
2955306	2022-09-20 19:05:00	1.086376e-06
2955307	2022-09-20 19:06:00	9.951779e-07
2955308	2022-09-20 19:07:00	9.110779e-07
2955309	2022-09-20 19:08:00	7.984809e-07
2955310	2022-09-20 19:09:00	7.122745e-07
2955311	2022-09-20 19:10:00	6.498038e-07
2955312	2022-09-20 19:11:00	6.001829e-07
2955313	2022-09-20 19:12:00	5.605073e-07
2955314	2022-09-20 19:13:00	5.235879e-07
2955315	2022-09-20 19:14:00	4.974534e-07
2955316	2022-09-20 19:15:00	4.719950e-07
2955317	2022-09-20 19:16:00	4.410630e-07
2955318	2022-09-20 19:17:00	4.099924e-07
2955319	2022-09-20 19:18:00	3.869208e-07


Earth Scale
AIA 304
2021-09-21 06:00:41
www.helioviewer.org


Earth Scale
AIA 171
2021-09-21 06:00:45
www.helioviewer.org


Earth Scale
HMI continuum
2021-09-21 06:01:18
www.helioviewer.org

Conclusion

Our model succeeded in identifying some flare events in the NASA catalog. However, we did not detect the majority of them.

Further development of machine learning models such as convolutional autoencoders holds promise for improving upon the incumbent flare detection computational algorithm.

Acknowledgments

Thank you for the opportunity to build our data science skills and assist in exciting real-world research!

Vinay Kashyap

Bernhard Klingenberg

Joshua Ingram