NH51C-0795

Using Deep Learning to Automate Inference of Meteoroid Pre-Entry Properties

Ana María Tárano [1,2] (ana.m.tarano@nasa.gov), Jonathan Gee [2,3], Lorien Wheeler [4], Sigrid Close [1], Donovan Mathias [4]
1. Aeronautics & Astronautics, Stanford University, Stanford, CA, United States. 2. Science and Technology Corp./NASA Ames, Moffett Field, CA, United States. 3. Computer Science, San Jose State University, San Jose, CA, United States. 4. NASA Ames Research Center, Moffett Field, CA, United States.

0. Abstract

Properly assessing the asteroid threat depends on the knowledge of asteroid pre-entry parameters, such as size, velocity, mass, density, and strength. Although a vast number of possible bodies to study exist, such characterization of asteroid populations is currently limited by substantial costs associated with space rendezvous missions and rare meteorite findings. As asteroids fragment, ablate, and decelerate in the atmosphere, they emit light detectable by ground-based and space-borne instruments. Earth's atmosphere, thus, becomes an accessible laboratory that enables impactor risk assessments by facilitating inference of the pre-entry parameters. These asteroid pre-entry conditions are typically deduced by modeling the entry and breakup physics that best reproduce the observed light or energy deposition curve. However, this process requires extensive manual trial-and-error of uncertain modeling parameters. Automating meteor modeling and inference would improve property distributions used in risk assessments and enable population characterization as more light curves become more readily available through the presence of space assets and ground-based camera networks.

We previously developed a genetic algorithm to automate meteor modeling by using the fragment-cloud model (FCM) to search for the values of the FCM input parameters (e.g., diameter) that generate energy deposition profiles that match the observed one. Now, we apply deep learning to infer asteroid diameter, velocity, and density from observed energy deposition curves. We trained and tested our neural network models with synthetic energy deposition curves modeled using the FCM rubble pile implementation. We present an application of a 1D convolutional neural network and compare its performance to other attempted regressors and machine learning techniques, such as a fully connected neural network and Random Forest regression, to demonstrate its capabilities. We validate our model weights and approach using the Chelyabinsk, Tagish Lake, Benešov, Košice, and Lost City meteors.

1. Motivation: Unlabeled but abundant data sets

A. Background

- Asteroids pose a threat to humanity and the environment.
- Every day, 80 to 100 tons of material falls upon Earth from space in the form of dust and small meteorites.
- Asteroid missions are expensive, and meteorites are rare.
- Studying meteoroids entering the atmosphere is becoming more readily available through the presence of space assets, such as the Geostationary Lightning Mapper, and growing ground-based camera networks.
- Pre-entry parameters, such as diameter, density, angle, velocity, and aerodynamic strength, are critical for asteroid threat assessment.
- The pre-entry parameters of impacting asteroids are not directly measured from energy deposition curves derived from optical sensors.
- Physics-based models and uncertain mean values are used to infer unknown quantities from energy deposition curves when velocity and entry angle are known.

B. Challenges

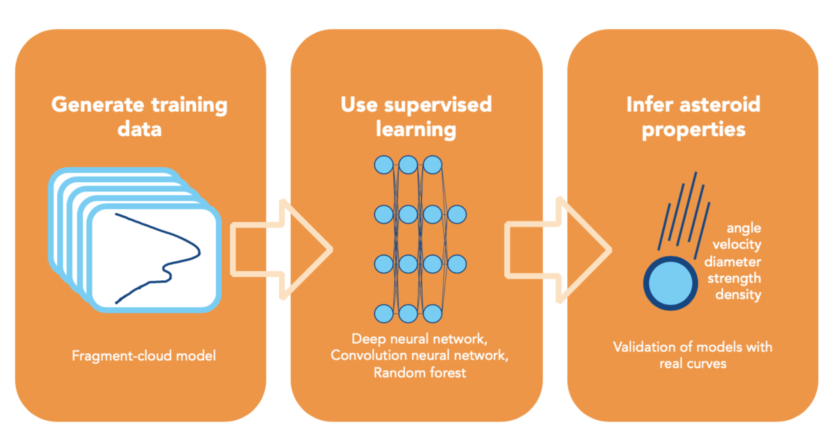
- Light curve data from asteroids entering the atmosphere is abundant but the asteroid's properties are not commonly directly observed so there is ample but incomplete and unlabeled data.
- The meteor and asteroid communities rely heavily on modeling to infer properties from the data by reproducing the energy signatures that were observed.
- We have previously used a genetic algorithm to reproduce the manual labor of curve matching to solve for model inputs using a semi-analytical fragment-cloud model (FCM).
- We leverage an extended version of FCM to generate labeled data to train regression models in order to infer model inputs from observed cases.

C. Science Objective

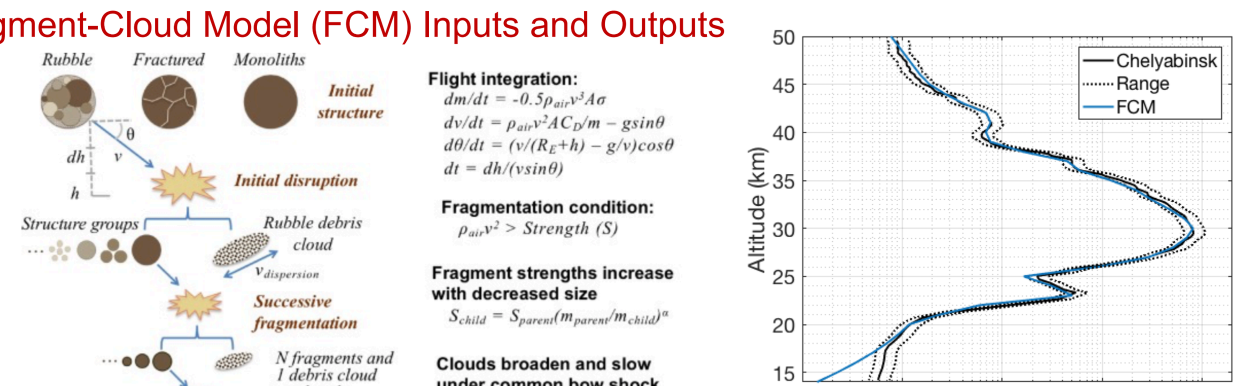
Can our synthetically trained regression models be generalized to infer parameters from real fireballs?

2. Methodology: Train using physics-based synthetic data

A. Overview of Process



B. Fragment-Cloud Model (FCM) Inputs and Outputs



Range and distribution of the input parameters generated to create synthetic FCM data sets to train the supervised learning models on.

Parameter	Minimum	Maximum	Distribution
Diameter (m)	0.1	50	Uniform
Velocity (km/s)	11	25	Uniform
Angle (°)	10	90	Uniform
Bulk Density (g/cm^3)	1.1	4.0	Uniform
Strength (kPa)	1.0	15,000	Log-uniform

3. Methodology: Top 3 regression models

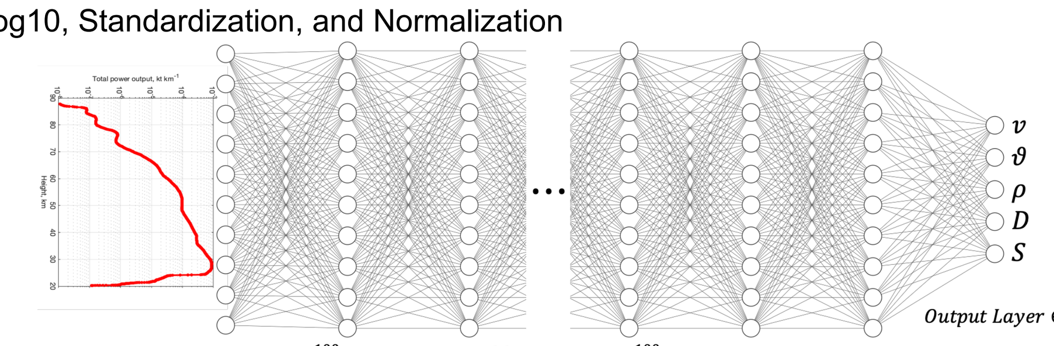
- Although we attempted the use of different regression models, such as gradient boost and support vector regression, and attempted many designs, not all models were able to train. Here we present the 3 most successful frameworks to enable training and similarly valued validation metrics.
- Our general strategy in deriving the presented topologies involved in optimizing for the maximization of the R^2 score of the predictions made to the validation data set, minimizing the loss of the real cases, and avoiding overfitting.

C. Model 1: Deep Neural Network (DNN)

Data Preparation:

Inputs:
- Altitude data → Log10, Standardization, and Normalization for all
- 100 energy deposition points

Outputs:
- Velocity → Standardization and Normalization
- Angle → Standardization and Normalization
  Density → Standardization and Normalization
  Diameter → Standardization and Normalization
- Strength → Log10, Standardization, and Normalization



Data Processing:

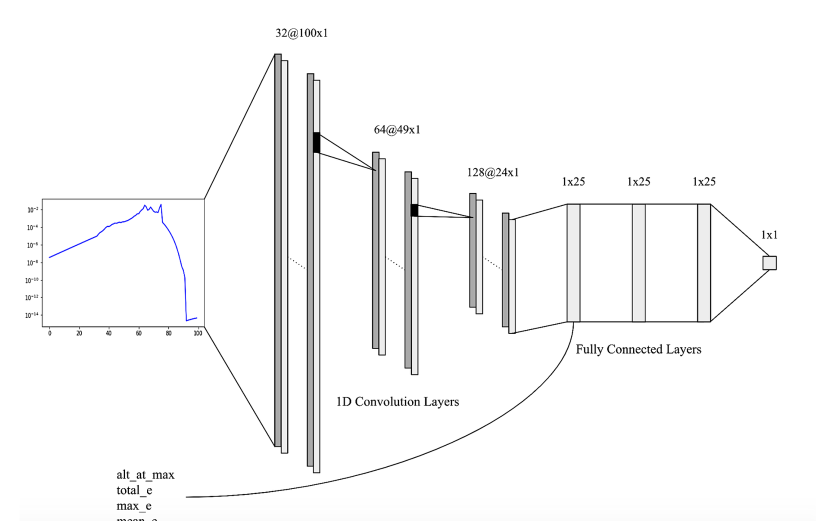
- Epochs: 150
- Loss Function: Mean-squared error (MSE)
- Solver: Adaptive Moment Estimation (Adam)
- Hidden Layers: 20
- Output Layer: 5 parameters
- Activation Functions: Rectified Linear Unit (ReLU) for all except output layer that uses linear
- Regularization: None
- Dropout: None

D. Model 2: Convolutional Neural Network (CNN)

Data Preparation:

Inputs:
- Altitude data → Log10 and Normalization for all
- 100 energy deposition points

Outputs:
- Velocity → Normalization
- Angle → Normalization
  Density → Normalization
  Diameter → Normalization
- Strength → Log10 and Normalization



Data Processing:

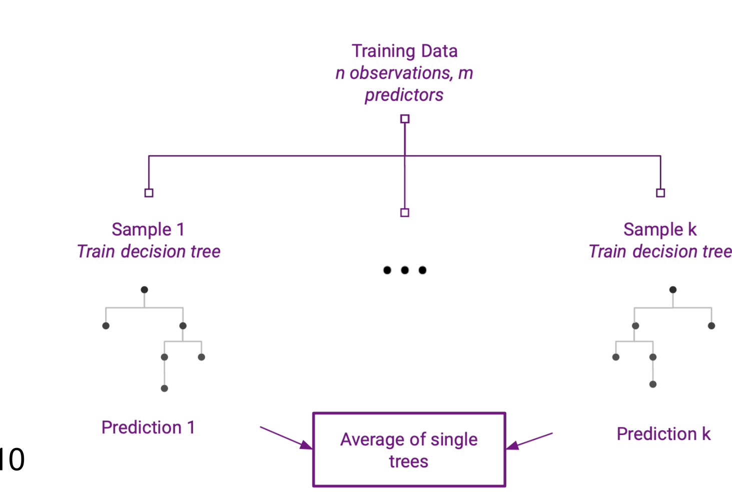
- Epochs: 50
- Loss Function: MSE
- Solver: Adam
- Convolutional Layers: 3 (1D)
- Hidden Layers: 3
  - Augmented feature space: total energy deposited, mean energy deposited, altitude at maximum energy deposition, and maximum energy deposition rate
  - Add 4 additional features
- Output Layer: 1 parameter
- Activation Functions: ReLU
- Regularization: None
- Dropout: None

E. Model 3: Random Forest Regression (RFR)

Data Preparation:

Inputs:
- Altitude data → different depending on the output
  - Raw for Angle and Diameter
  - Log10 for Velocity, Density, and Strength
- Augmented feature space: total energy deposited, mean energy deposited, altitude at maximum energy deposition, and maximum energy deposition rate
- 104: 100 energy deposition points and 4 additional features

Outputs:
- Velocity
- Angle
- Density
- Diameter
- Strength → Log10

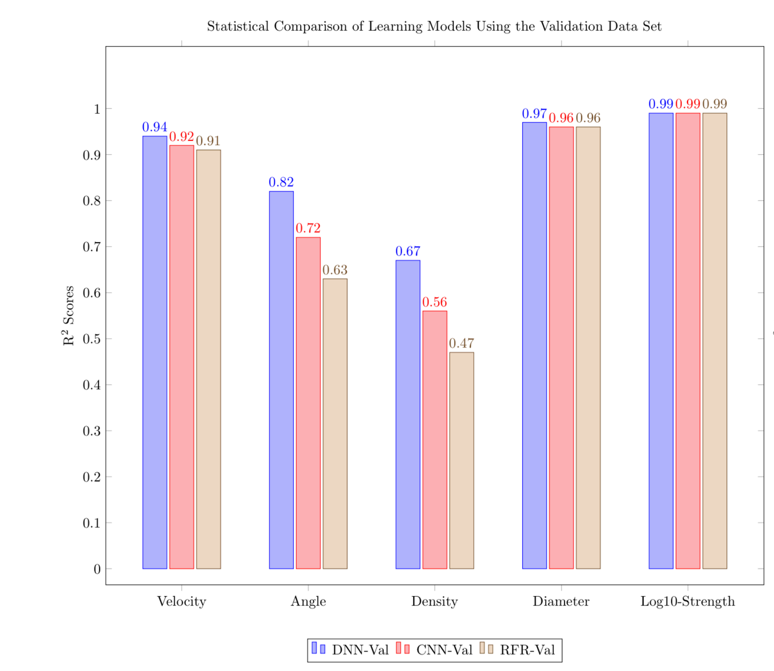


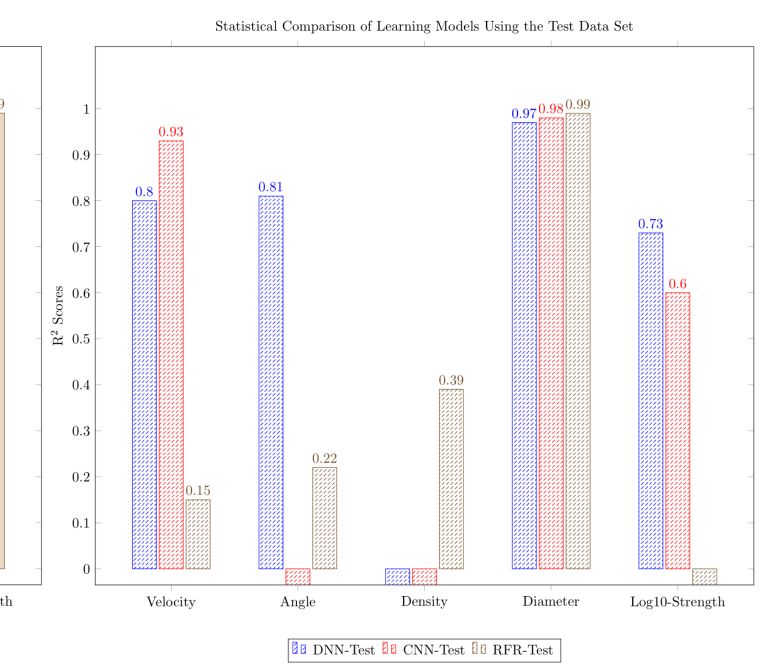
Data Processing:

- Trees:
  - Velocity: 5
  - Angle, Density, Diameter, and Strength: 10
- Output Layer: 1 parameter

4. Results: DNN provides best generalization overall

- In order to compare the different methods' ability to predict unseen data, we use the **R^2 score** metric, also known as the coefficient of determination. For the *scikit-learn* package, the best possible score is 1.0 and the score can be negative. A score of 0 indicates that the mean is much better predictor than the model used.
- We present the R^2 scores for each output parameter for the validation and test sets. The **validation R^2 scores** highlight whether the models can predict unseen data that is like the training set. The **test R^2 scores** will provide insight as to whether the models are generalizable to real observed data.





- The scores indicate that the DNN is the most generalizable approach overall for both the synthetically-generated and realistic curves, excluding the density target.
- If we focus on each individual modeling parameter, then the CNN outperforms the other approaches for inferring velocity. The DNN exceeds the other methods for inferring angle and strength. The RFR is most adept at generalizing solutions for density. Diameter can be inferred consistently by either method.

Velocity

Test Case	Actual (km/s)	Prediction (km/s)	Error (km/s)	Percentage Error (%)
Deep Neural Network				
Lost City	14.2	13.7	-0.50	-3.52
Benesov	21.5	23.4	1.90	8.84
Tagish Lake	15.8	16.8	1.00	6.33
Kosice	15.0	14.2	-0.80	-5.33
Chelyabinsk	19.2	20.7	1.50	7.81
Convolutional Neural Network				
Lost City	14.2	13.7	-0.50	-3.52
Benesov	21.5	20.0	-1.50	-6.98
Tagish Lake	15.8	16.3	0.50	3.16
Kosice	15.0	15.0	0.00	0.00
Chelyabinsk	19.2	19.2	0.00	0.00
Random Forest Regression				
Lost City	14.2	11.9	-2.30	-16.2
Benesov	21.5	22.4	0.90	4.19
Tagish Lake	15.8	20.7	4.90	31.0
Kosice	15.0	15.1	0.10	0.67
Chelyabinsk	19.2	20.8	1.60	8.33

Angle

Test Case	Actual (°)	Prediction (°)	Error (°)	Percentage Error (%)
Deep Neural Network				
Lost City	38.0	50.9	12.9	33.9
Benesov	81.0	77.1	-3.90	-4.81
Tagish Lake	17.8	22.4	4.60	25.8
Kosice	60.0	41.1	-18.9	-31.5
Chelyabinsk	18.3	17.9	-0.40	-2.19
Convolutional Neural Network				
Lost City	38.0	57.0	19.0	50.0
Benesov	81.0	78.3	-2.70	-3.33
Tagish Lake	17.8	68.9	51.1	287
Kosice	60.0	74.5	14.5	24.2
Chelyabinsk	18.3	23.5	5.20	28.4
Random Forest Regression				
Lost City	38.0	53.5	15.5	40.8
Benesov	81.0	51.5	-29.5	-36.4
Tagish Lake	17.8	19.7	1.90	10.7
Kosice	60.0	27.0	-33.0	-55.0
Chelyabinsk	18.3	29.1	10.8	59.0

Density

Test Case	Actual (g/cm^3)	Prediction (g/cm^3)	Error (g/cm^3)	Percentage Error (%)
Deep Neural Network				
Lost City	3.40	1.54	-1.86	-54.7
Benesov	3.20	3.58	0.38	11.9
Tagish Lake	1.64	1.93	0.29	17.7
Kosice	2.50	1.54	-0.96	-38.4
Chelyabinsk	2.50	3.48	0.98	39.2
Convolutional Neural Network				
Lost City	3.40	2.01	-1.39	-40.9
Benesov	3.20	2.94	-0.26	-8.13
Tagish Lake	1.64	1.48	-0.16	-9.76
Kosice	2.50	3.32	0.82	32.8
Chelyabinsk	2.50	2.07	-0.43	-17.2
Random Forest Regression				
Lost City	3.40	3.42	0.02	0.59
Benesov	3.20	3.47	0.27	8.44
Tagish Lake	1.64	1.76	0.12	7.32
Kosice	2.50	3.53	1.03	41.2
Chelyabinsk	2.50	2.31	-0.19	-7.60

Diameter

Test Case	Actual (m)	Prediction (m)	Error (m)	Percentage Error (%)
Deep Neural Network				
Lost City	0.45	1.47	1.02	227
Benesov	1.35	0.83	-0.52	-38.5
Tagish Lake	4.50	5.56	1.06	23.6
Kosice	1.39	2.44	1.05	75.5
Chelyabinsk	19.8	17.5	-2.30	-11.6
Convolutional Neural Network				
Lost City	0.45	0.65	0.20	44.4
Benesov	1.35	1.89	0.54	21.5
Tagish Lake	4.50	5.01	0.51	11.3
Kosice	1.39	1.81	0.42	30.2
Chelyabinsk	19.8	21.8	2.00	10.1
Random Forest Regression				
Lost City	0.45	0.52	0.07	15.6
Benesov	1.35	1.64	0.29	40.0
Tagish Lake	4.50	3.31	-1.19	-26.4
Kosice	1.39	1.51	0.12	8.63
Chelyabinsk	19.8	19.3	-0.50	-2.53

Aerodynamic Strength

Test Case	Actual (kPa)	Prediction (kPa)	Error (kPa)	Percentage Error (%)
Deep Neural Network				
Lost City	0.25	4.22	3.97	1,590
Benesov	20.0	46.1	26.1	131
Tagish Lake	0.50	1.91	1.41	282
Kosice	2.00	2.97	0.97	48.5
Chelyabinsk	600	1030	430	71.7
Convolutional Neural Network				
Lost City	0.25	6.64	6.39	2,560
Benesov	20.0	32.9	12.9	64.5
Tagish Lake	0.50	4.49	3.99	798
Kosice	2.00	2.61	0.61	30.5
Chelyabinsk	600	679	79.0	13.2
Random Forest Regression				
Lost City	0.25	257	257	103,000
Benesov	20.0	94.4	74.4	372
Tagish Lake	0.50	2.33	1.83	366
Kosice	2.00	19.2	17.2	860
Chelyabinsk	600	634	34.0	5.67

- Tables above demonstrate the range of error and the relative error for each method by output parameter.

Diameter	Density	Velocity	Angle wrt Horizontal
18.6%	**13.0%**	**2.73%**	**19.6%**
Mean Absolute Percentage Error	Mean Absolute Percentage Error	Mean Absolute Percentage Error	Mean Absolute Percentage Error

5. Discussion

Logarithmic Transformations

- Transforming the input feature space enabled better predictions of velocity, angle, and density target variables, while it did not significantly improve predictions for diameter.
- Transforming strength enabled its inference because it reduced the wide-ranging target variable.

Strengths

- Excluding density, the DNN model should be used as a global approach for getting solutions for all the target variables needed.
- We have demonstrated a new capability of being able to deduce entry angle and velocity without requiring the use of explicit dynamic or trajectory information. This ability is highly sought after for observed light curves where <3 stations were able to observe and event or when sensors are less than 100 km apart.

Limitations

- Even though our training and validation sets didn't have smaller values of strength than 1 kPa, our DNN inferred small values for Lost City and Tagish Lake. The smallest strength values of the test set are not represented in the training data and performance could be improved by expanding training data set.
- High R^2 scores for the validation set do not always imply generalizability of the model to real data, as seen by the model's performances in for the density parameter.

Future Work

01 | Apply models to 10 potential other curves.
02 | Use sensor fusion to develop more robust systems.
03 | Develop methodology for incomplete data.
04 | Train models on synthetically generated light curves directly.

Acknowledgements and References

This work was funded by the NASA Planetary Defense Coordination Office. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center. TensorFlow and Keras were used to model the neural networks.

References

Avramenko et al. *J Geophysical Research: Atmospheres*. **2014**, 119. Borovička and Berezhnoy. *Icarus*. **2016**, 278. Borovička and Spurny. *Icarus*. **1996**, 121. Borovička et al. *Astronomy and Astrophysics*. **1998**, 334. Borovička et al. *Nature*. **2013**, 503. Borovička et al. *Asteroids IV*. **2015**, 257-280. Brown et al. *Science*. **2000**, 290. Brown et al. *Meteoritics and Planetary Science*. **2002**, 37. Brown et al. *Icarus*. **2016**, 266. Ceplecha and ReVelle. *Meteoritics & Planetary Science*. **2005**, 40. Gritsevich. *Solar System Research*. **2008**, 42. Hildebrand et al. *Meteoritics and Planetary Science*. **2006**, 41. Mathias et al. *Icarus*. **2017**, 289. McCrosky et al. *Journal of Geophysical Research*. **1971**, 76. Popova et al. *Meteoritics & Planetary Science*. **2011**, 46. Popova e t al. *Science*. **2013**, 342. Tárano et al. *Icarus*. **2019**, 329. Wheeler et al. *Icarus*. **2017**, 295. Wheeler et al. *Icarus*. **2018**, 315. Abadi et al., TensorFlow: Large-scale machine learning on heterogeneous systems, (tensorflow.org), 2015. Chollet et al., Keras, https://keras.io, 2015.