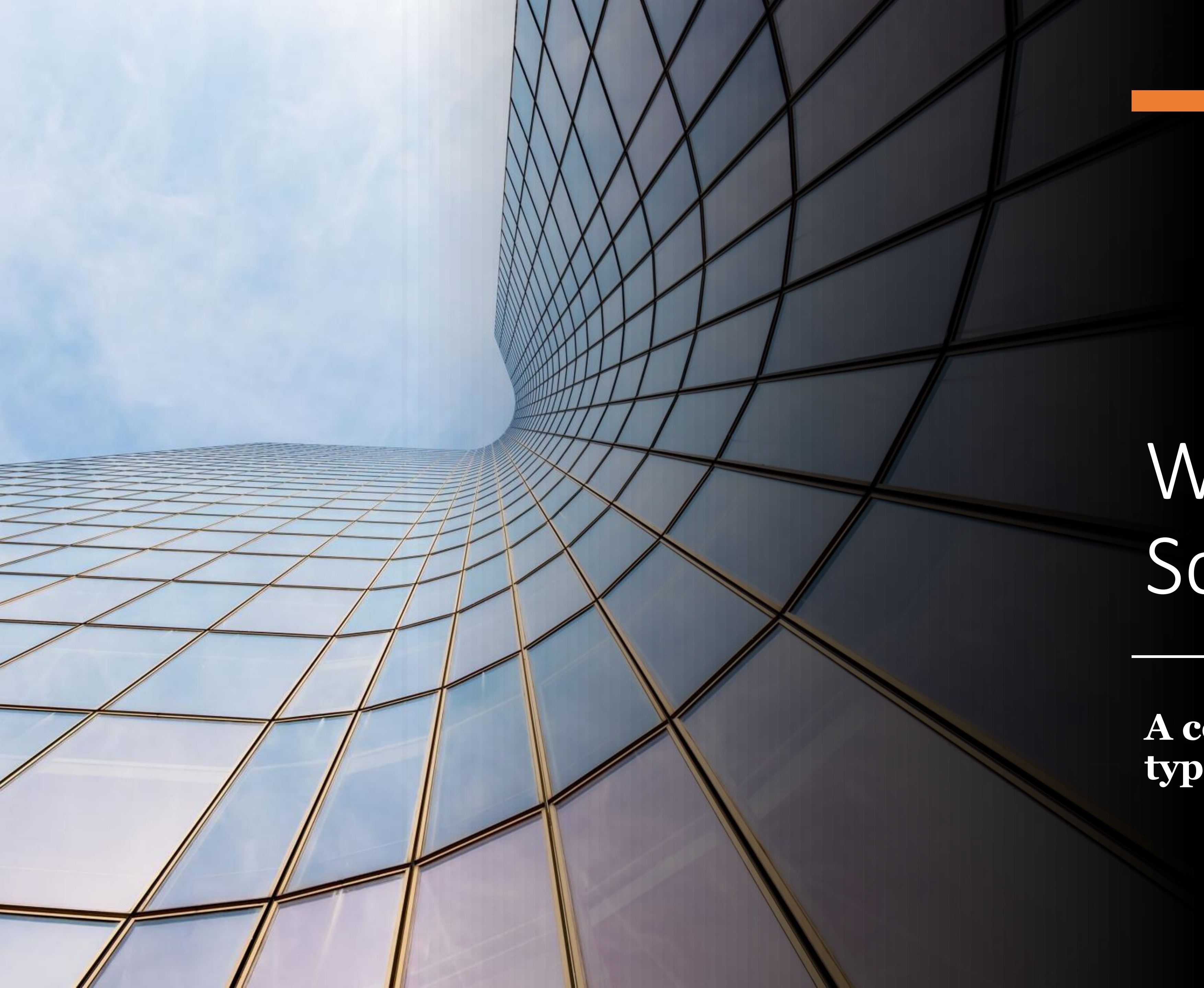
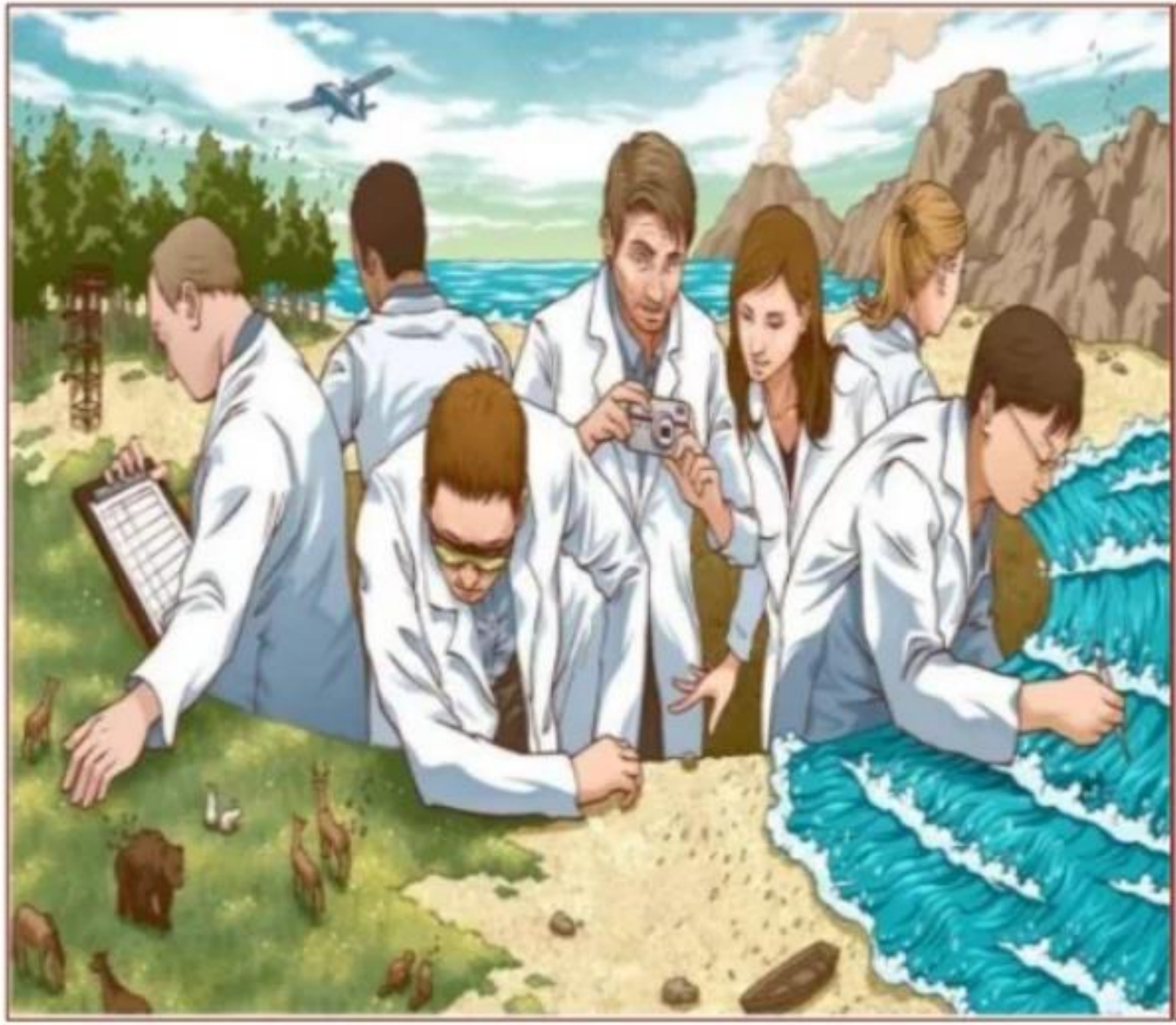


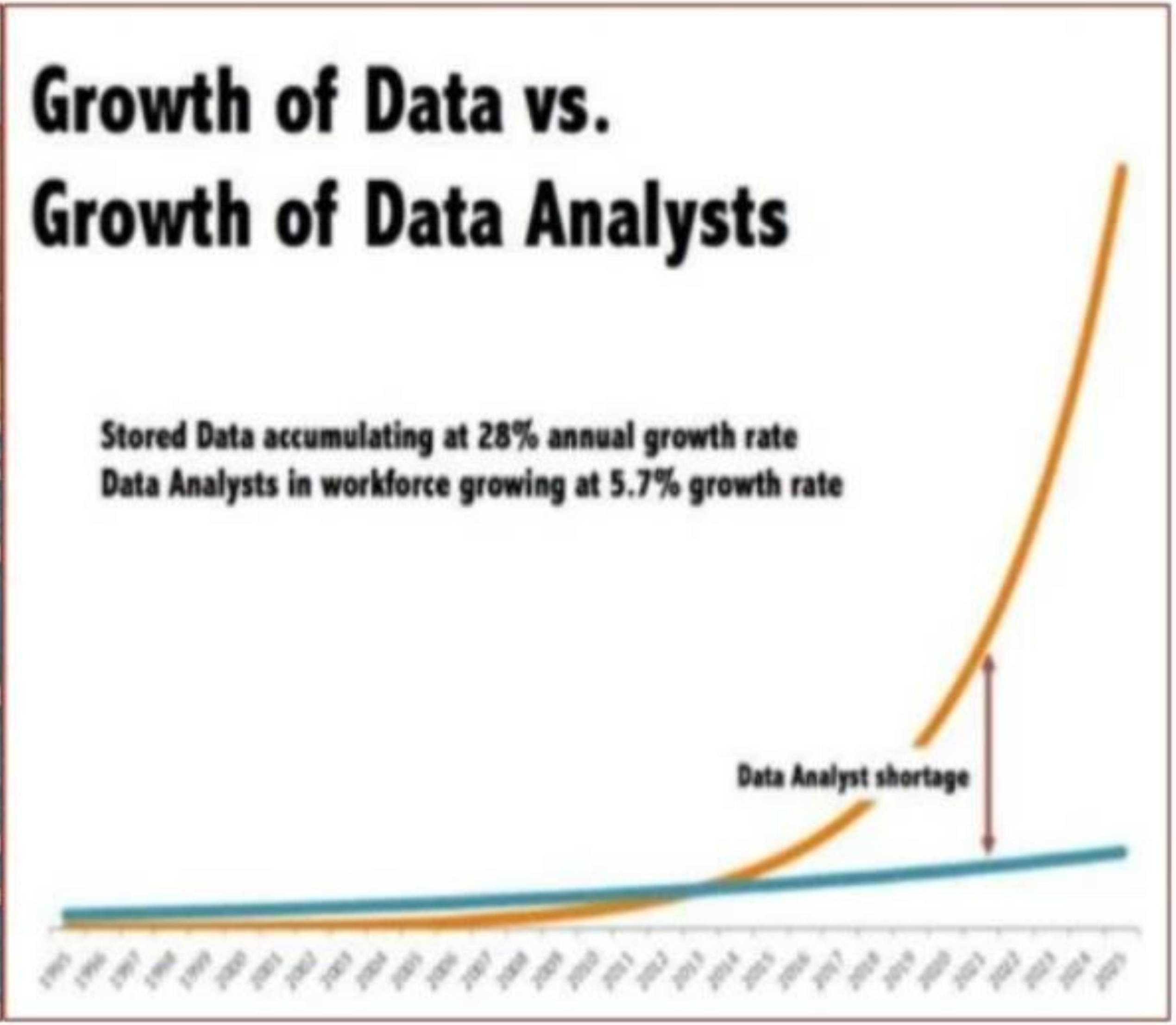
What is Data Science

A concise introduction with typical applications

What is Data Science - Outline

- What is Data Science
- Doing Data Science
- Data Science in Science
- Applications to Geomagnetic field
- Geomagnetic field moment predictions
- Core-Mantle Boundary (CMB) flow and Neural Networks
- Some conclusions
- References





Source: https://bit.ly/31HBHuQ

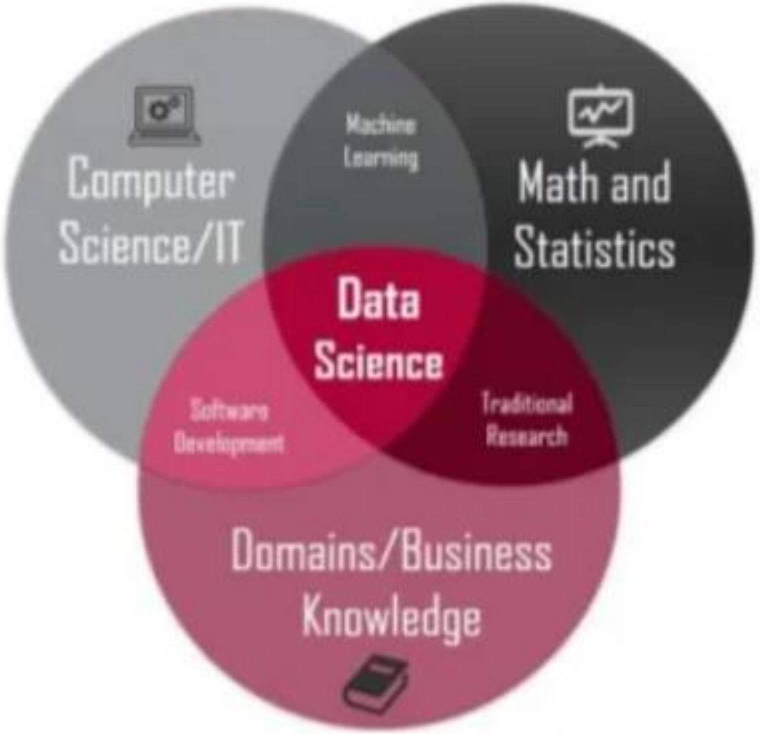
What is Data Science

- a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.

Source: https://bit.ly/30dekJB

What is Data Science

- a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data.
- employs techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, and information science.



Source: https://bit.ly/2YTRQ3w

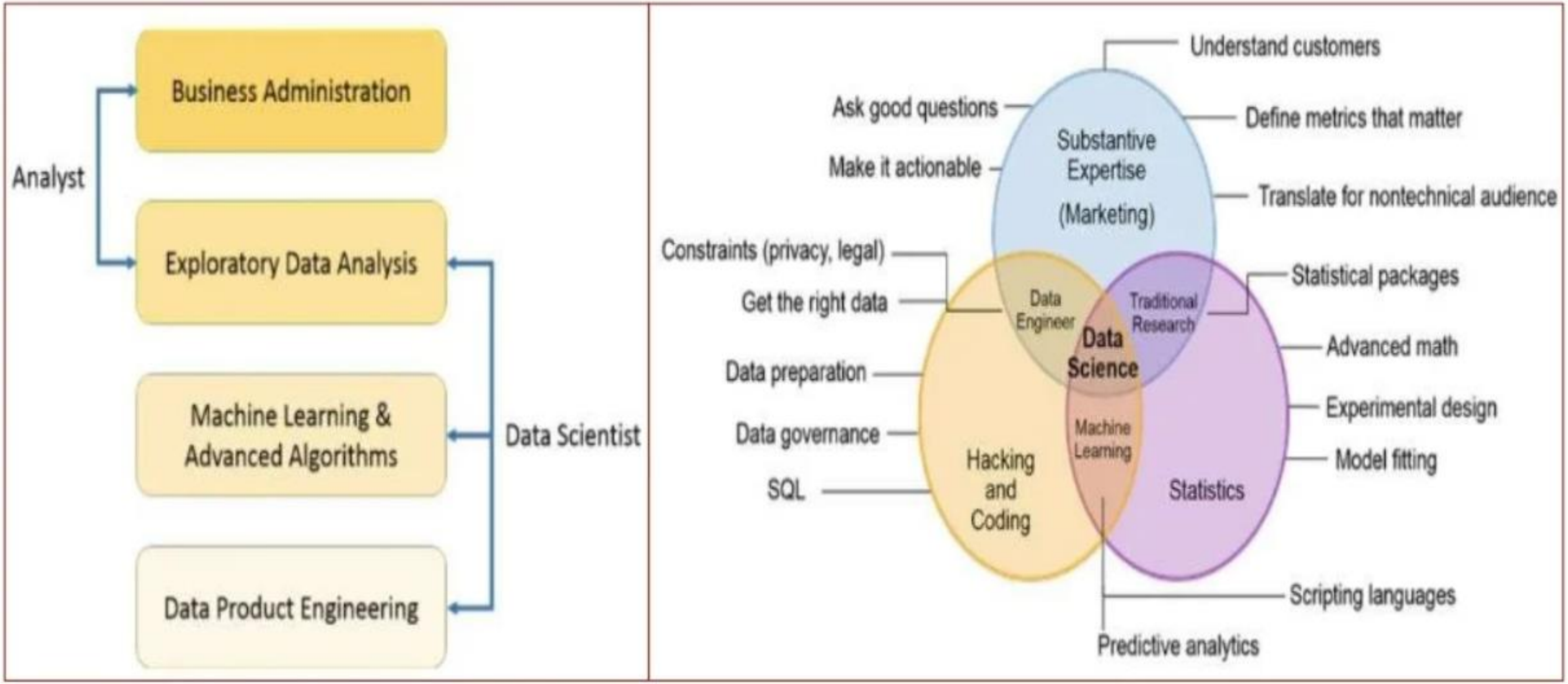
What is Data Science

"No candy? No flowers? No cards? Big Data predicted that 67.53% of you would remember!"

"I don't like the look of this. Searches for gravy and turkey stuffing are going through the roof !"

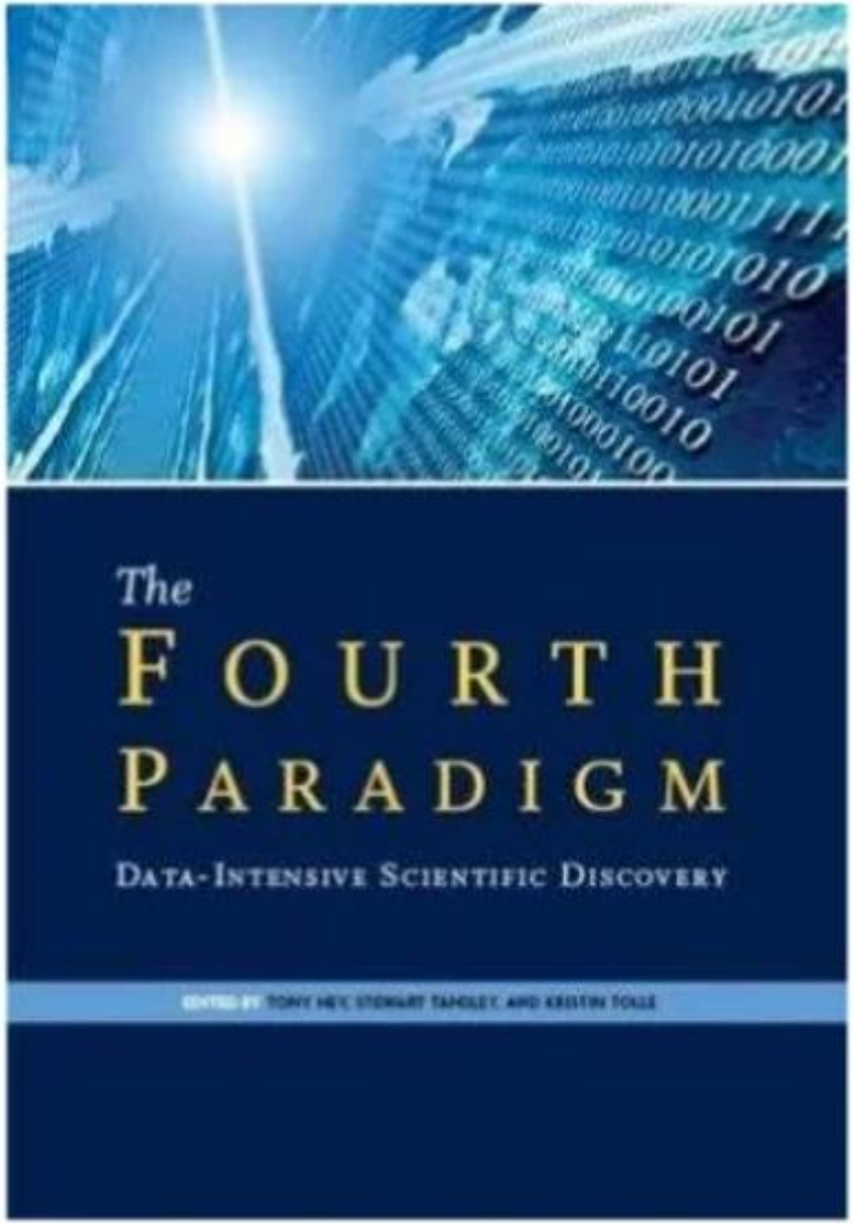
"You can't keep adjusting the data to prove that you would be the best Valentine's date for Scarlett Johansson."

What is Data Science

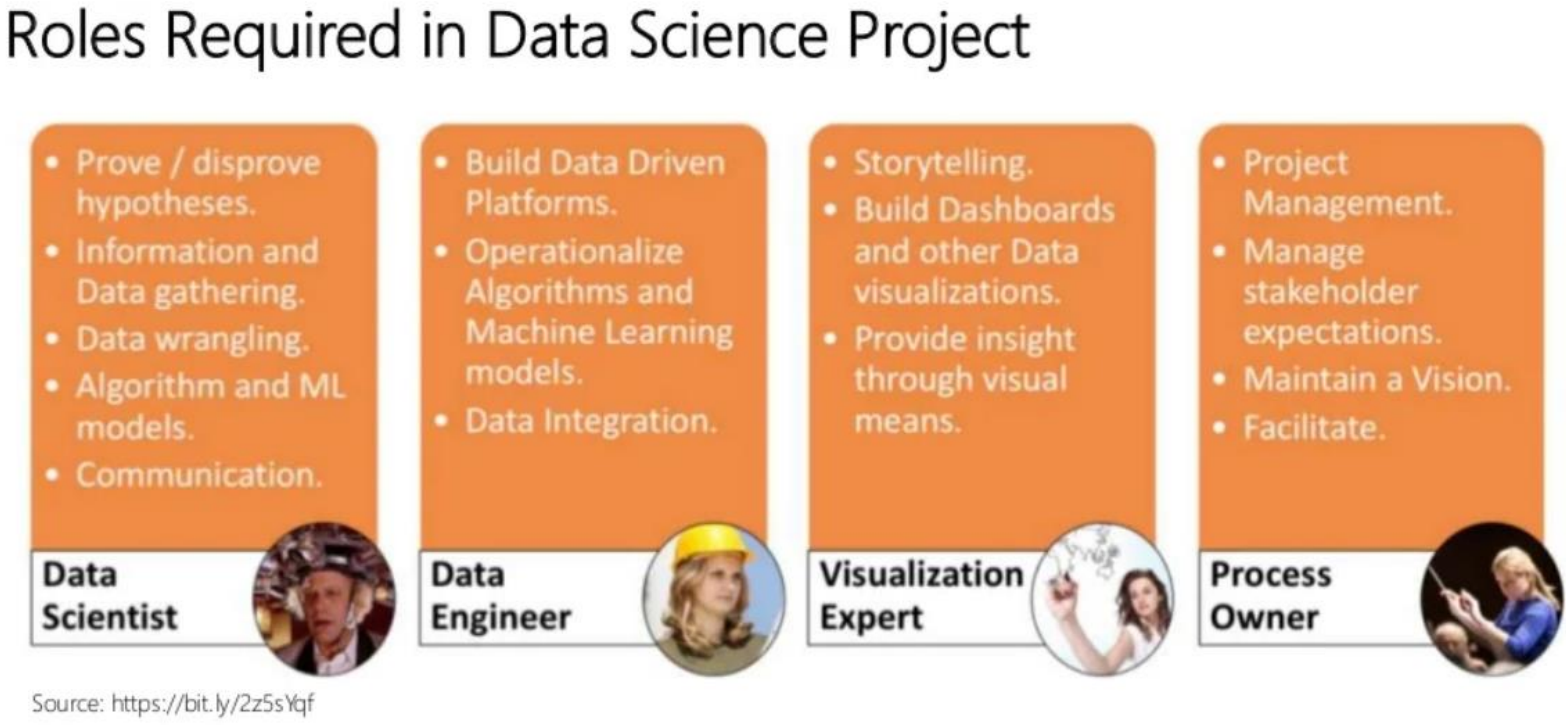


Fourth Paradigm of Science

- Thousand of years
  - Empirical
- Few hundred of years
  - Theoretical
- Last fifty years
  - Computational
  - "Query the world"
- Last twenty years
  - eScience (Data Science)
  - "Download the world"



What is Data Science



Roles Required in Data Science Project

- Prove / disprove hypotheses.
- Information and Data gathering.
- Data wrangling.
- Algorithm and ML models.
- Communication.

Data Scientist

- Build Data Driven Platforms.
- Operationalize Algorithms and Machine Learning models.
- Data Integration.

Data Engineer

- Storytelling.
- Build Dashboards and other Data visualizations.
- Provide insight through visual means.

Visualization Expert

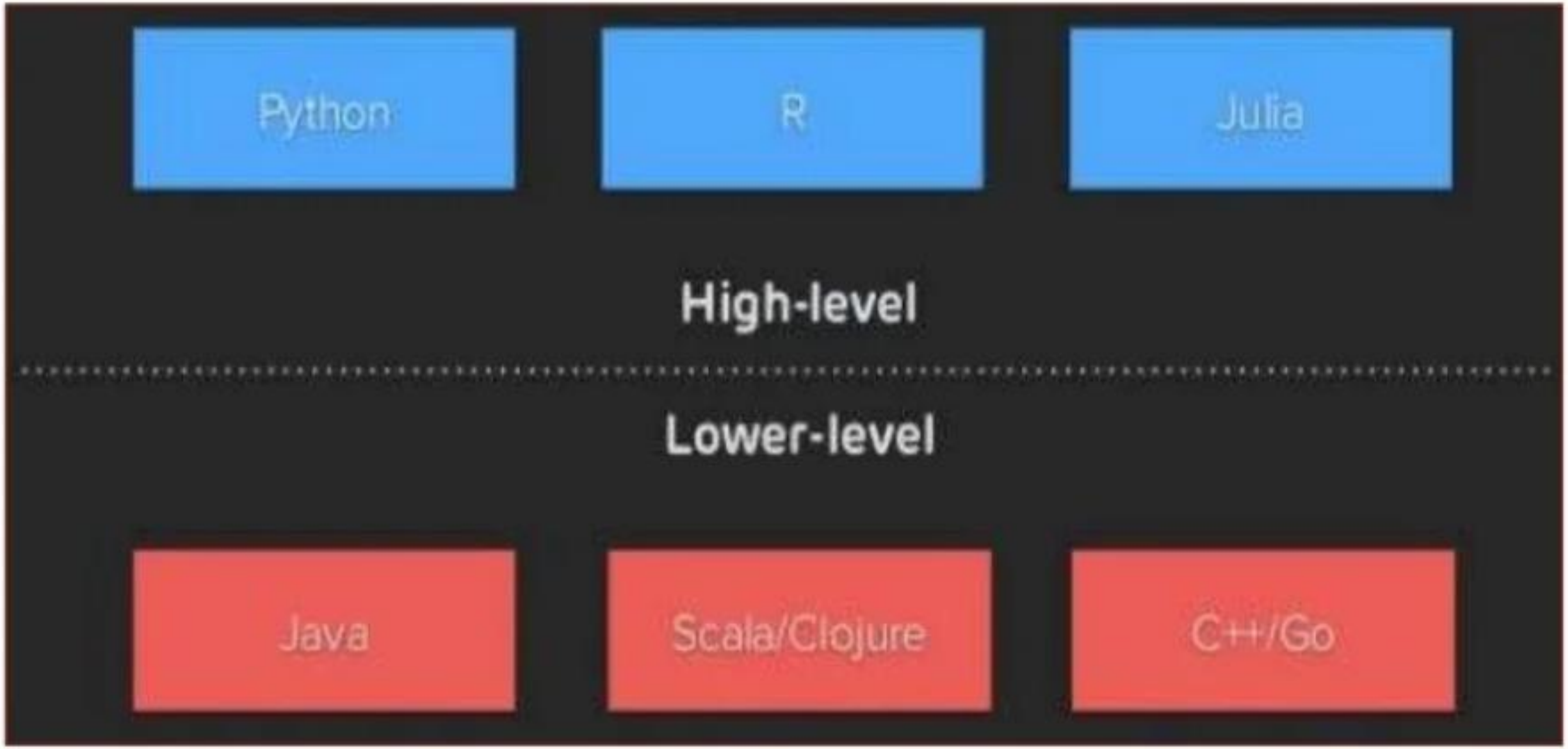
- Project Management.
- Manage stakeholder expectations.
- Maintain a Vision.
- Facilitate.

Process Owner

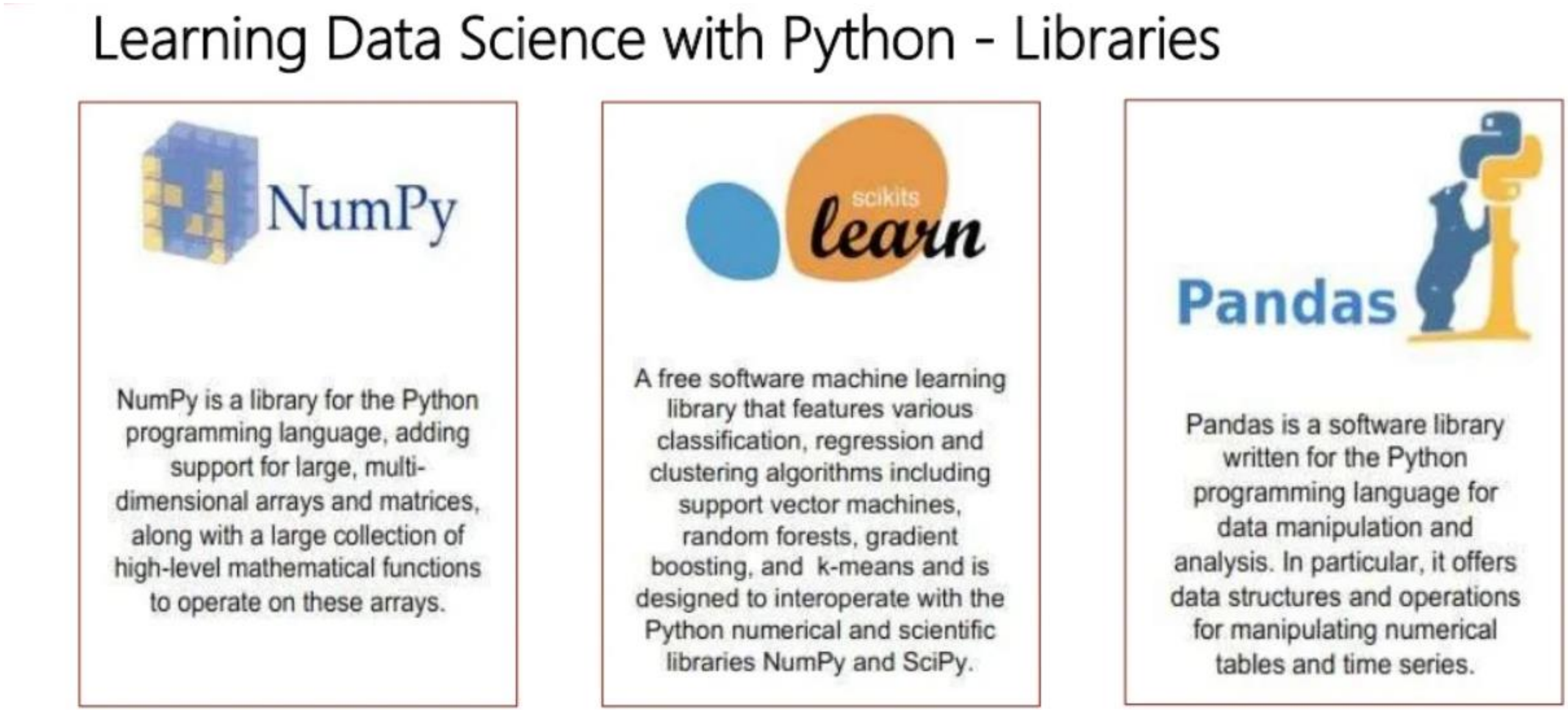
Source: https://bit.ly/2z5sYqf

How to become a data scientist?

- Data Scientists need to know how to "CODE"



What is Data Science



Learning Data Science with Python - Libraries

NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

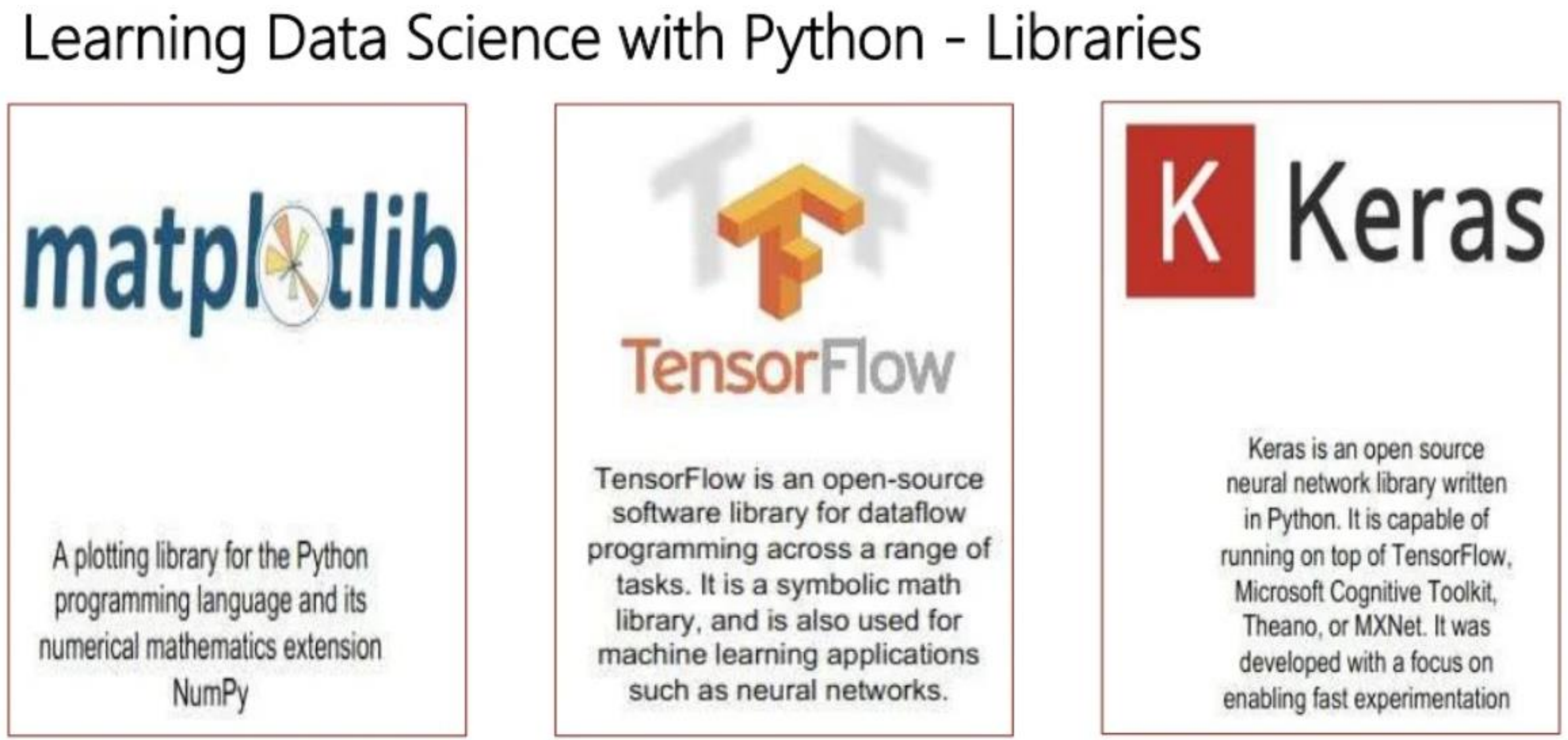
scikits learn

A free software machine learning library that features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, and k-means and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

What is Data Science



Learning Data Science with Python - Libraries

matplotlib

A plotting library for the Python programming language and its numerical mathematics extension NumPy

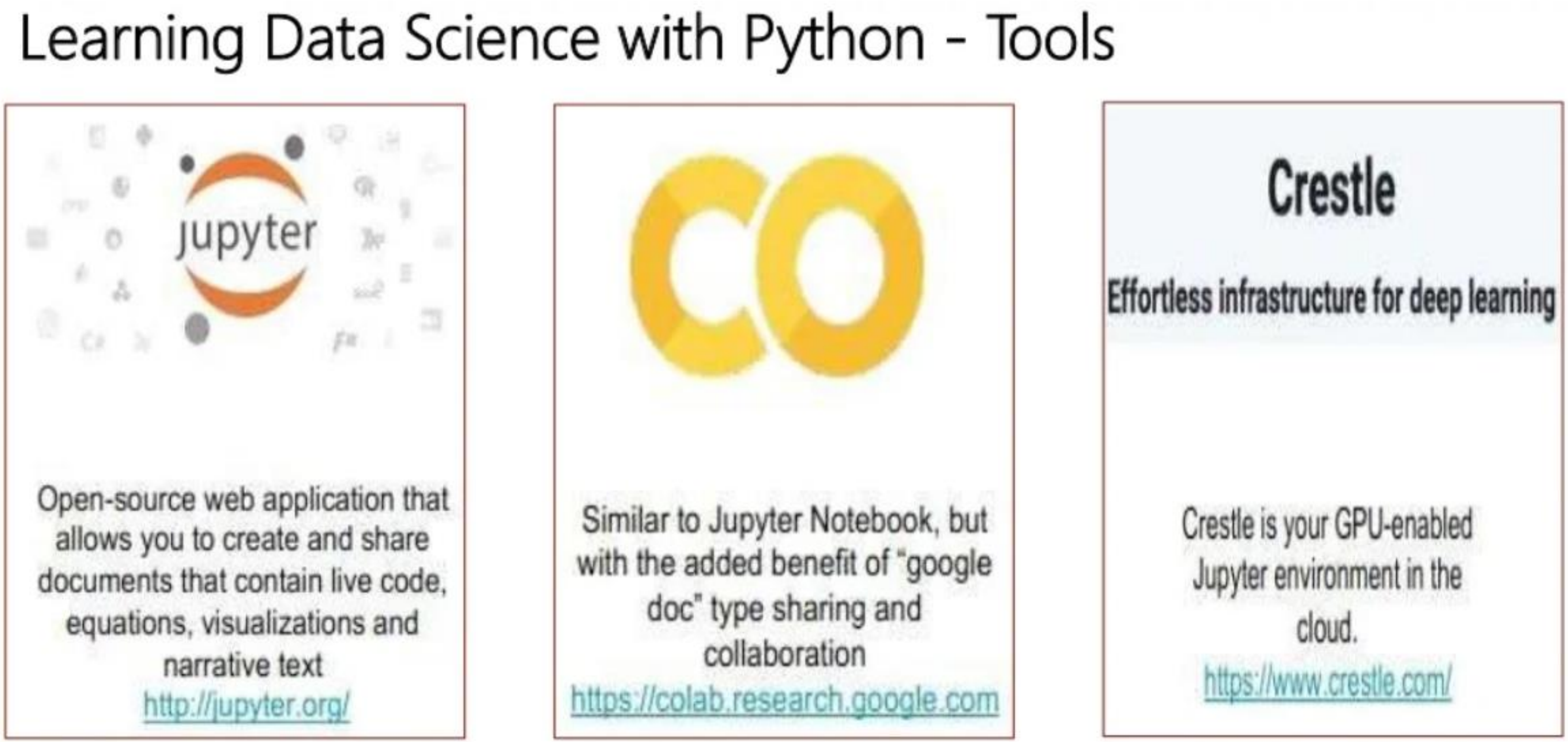
TensorFlow

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

Keras

Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or MXNet. It was developed with a focus on enabling fast experimentation

What is Data Science



Learning Data Science with Python - Tools

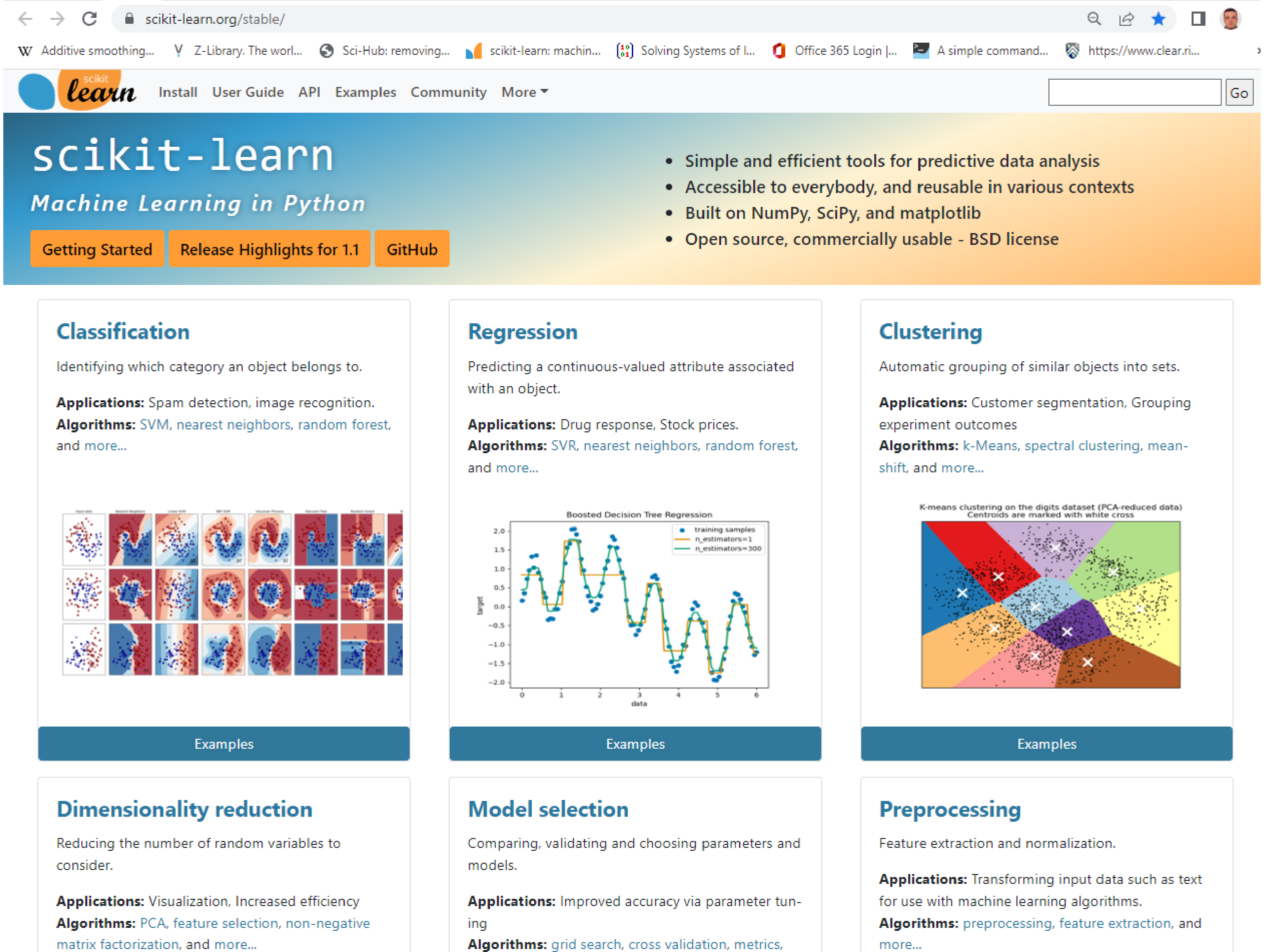
Open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text
http://jupyter.org/

Similar to Jupyter Notebook, but with the added benefit of "google doc" type sharing and collaboration
https://colab.research.google.com

Crestle

Effortless infrastructure for deep learning

Crestle is your GPU-enabled Jupyter environment in the cloud.
https://www.crestle.com/



scikit-learn

Machine Learning in Python

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

Model selection

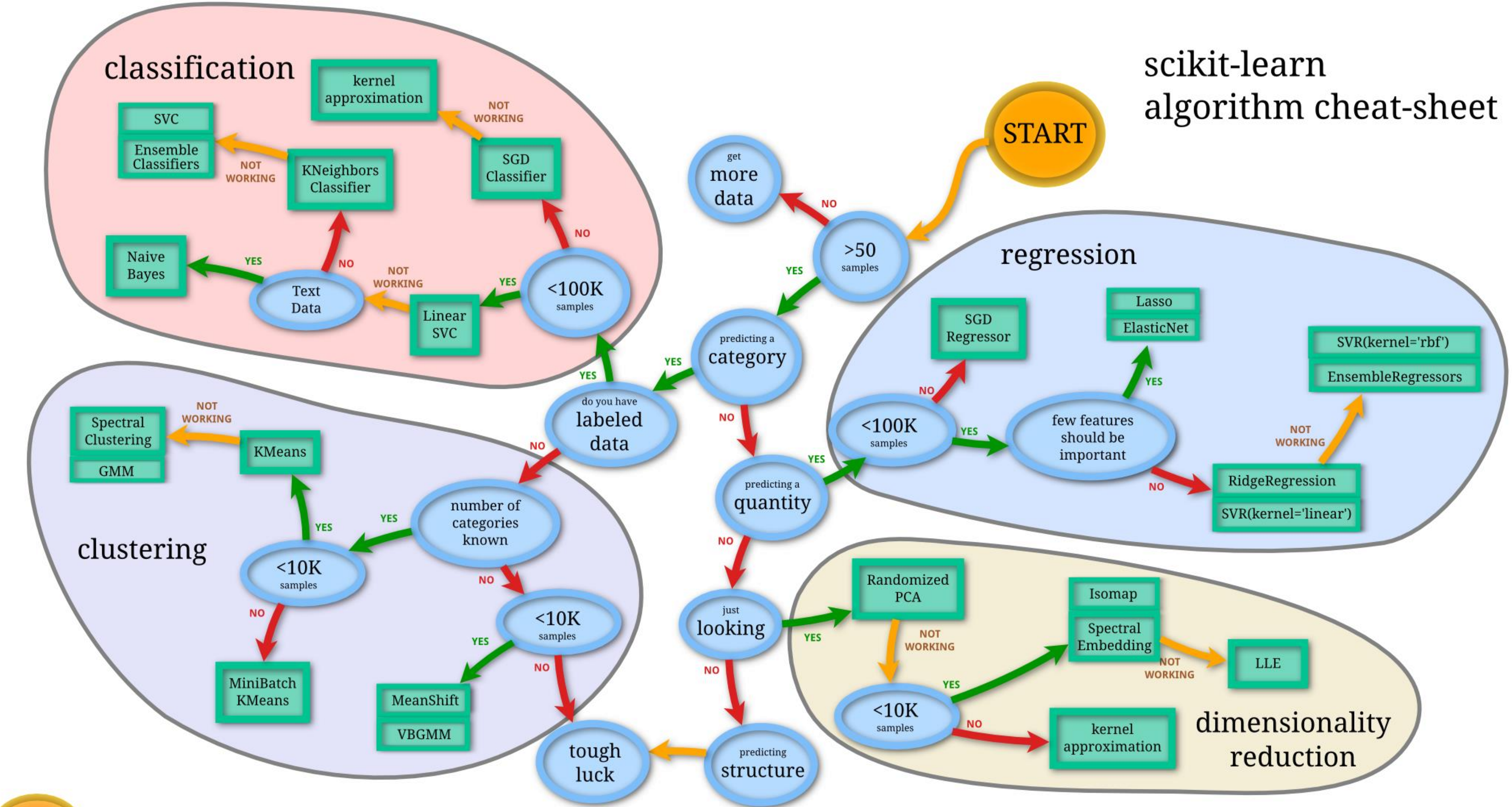
Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics,

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...



Back

scikit learn

Sci-kit cheat sheet

Geomagnetic field applications

- Background

Geomagnetic field applications

- Background
- Let discuss two typical examples involving:
    - Time series
    - Flow patterns

Geomagnetic field applications

- Background
- Let discuss two typical examples involving:
  - Time series
  - Flow patterns

- Don’t worry about the physical and mathematical details because you can apply the same approaches on a multitude of systems!

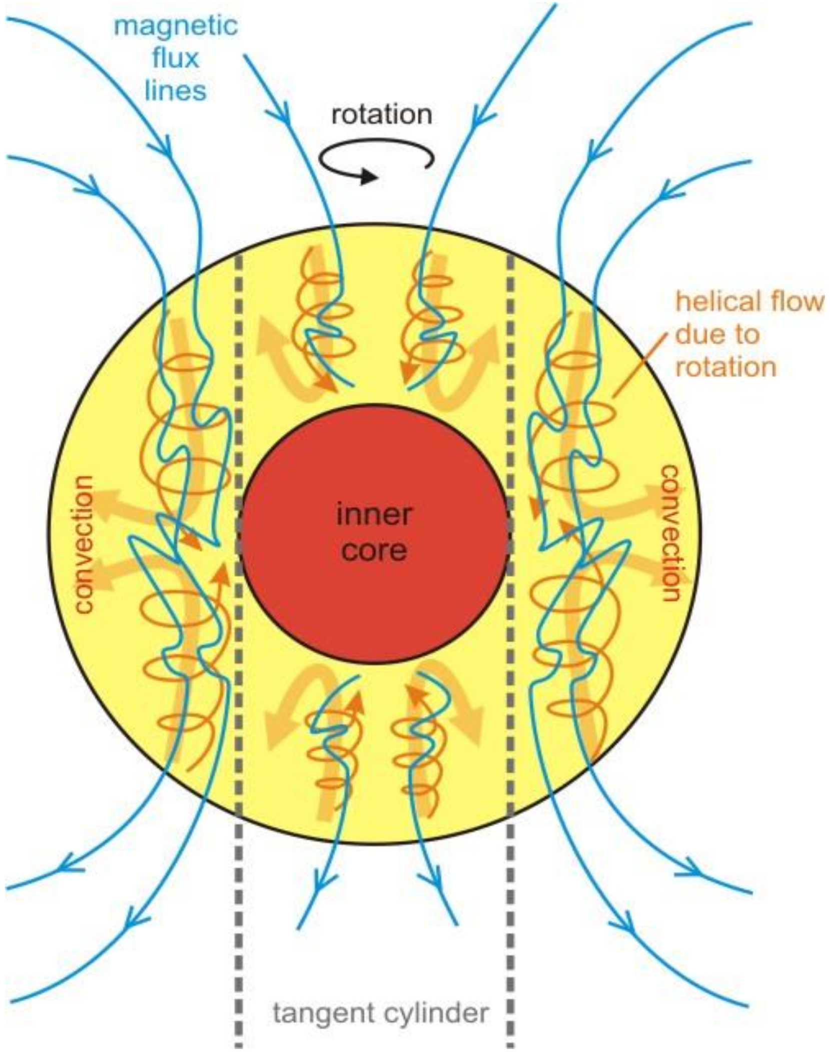
Background

- The Earth's main magnetic field is generated and maintained against Ohmic loss by dynamo mechanism. This mechanism takes place in the outer core. One of the main equation

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right) + \eta \nabla^2 \vec{B}$$

where the magnetic diffusivity $\eta = \frac{1}{\mu_0 \sigma}$, σ is electrical conductivity

- The geomagnetic field exhibits temporal variation on different timescales: from fraction of a second to millions of years.



Background

- The Earth's main magnetic field is generated and maintained against Ohmic loss by dynamo mechanism. This mechanism takes place in the outer core. One of the main equation

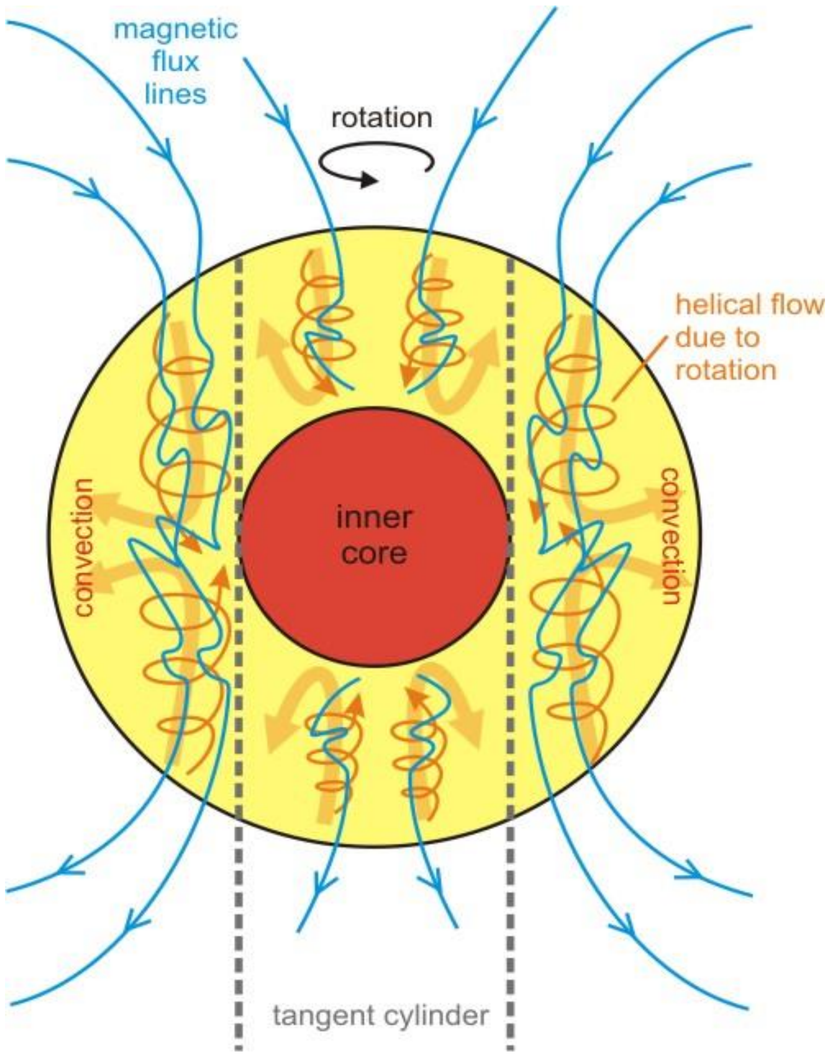
$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right) + \eta \nabla^2 \vec{B}$$

Source term

Dissipative term

where the magnetic diffusivity $\eta = \frac{1}{\mu_0 \sigma}$, σ is electrical conductivity

- The geomagnetic field exhibits temporal variation on different timescales: from fraction of a second to millions of years.



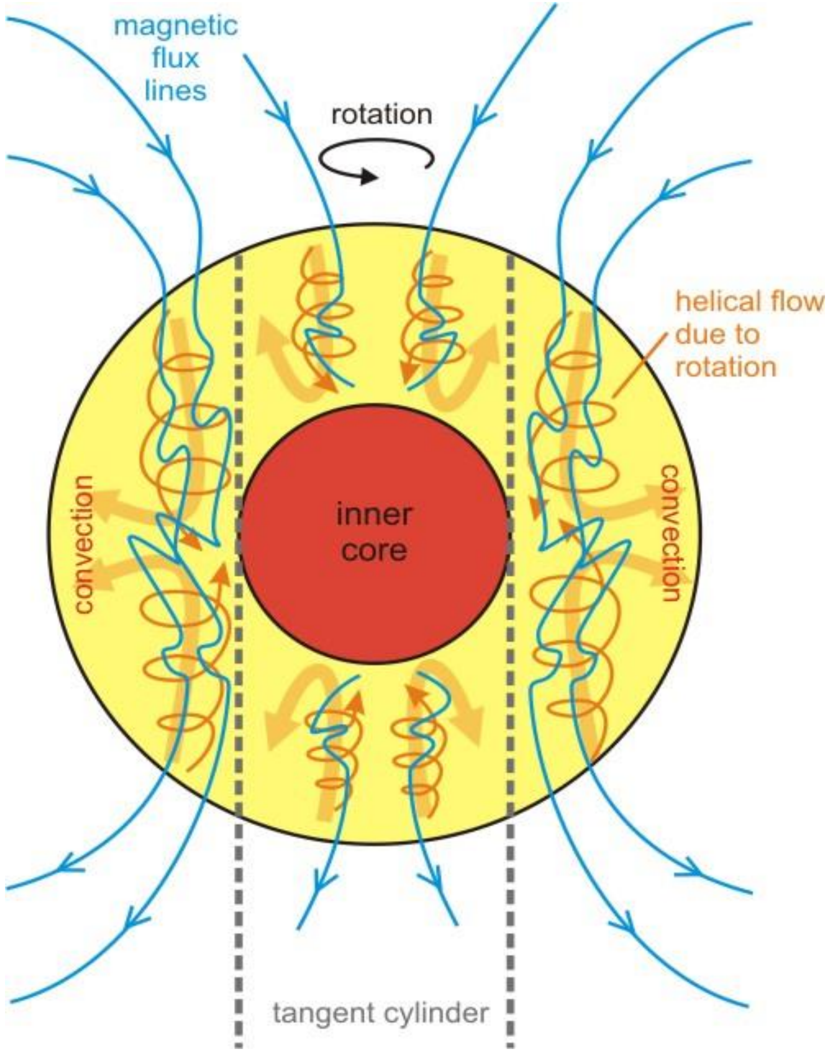
Background

- The Earth’s main magnetic field is generated and maintained against Ohmic loss by dynamo mechanism. This mechanism takes place in the outer core. One of the main equation

$$\underbrace{\frac{\partial \vec{B}}{\partial t}}_{\textbf{SV}} = \underbrace{\nabla \times \left(\vec{v} \times \vec{B}\right)}_{\textbf{Source term}} + \underbrace{\eta \nabla^2 \vec{B}}_{\textbf{Dissipative term}}$$

where the magnetic diffusivity $\eta = \frac{1}{\mu_0 \sigma}$, σ is electrical conductivity

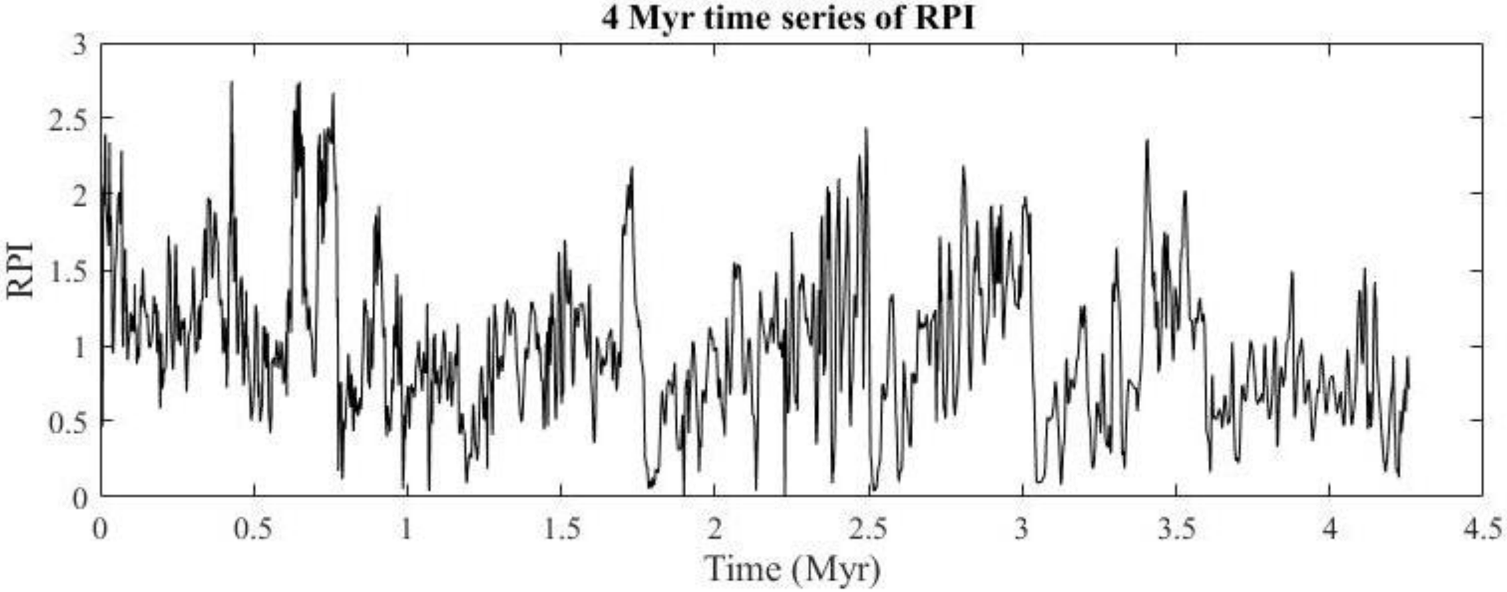
- The geomagnetic field exhibits temporal variation on different timescales: from fraction of a second to millions of years.

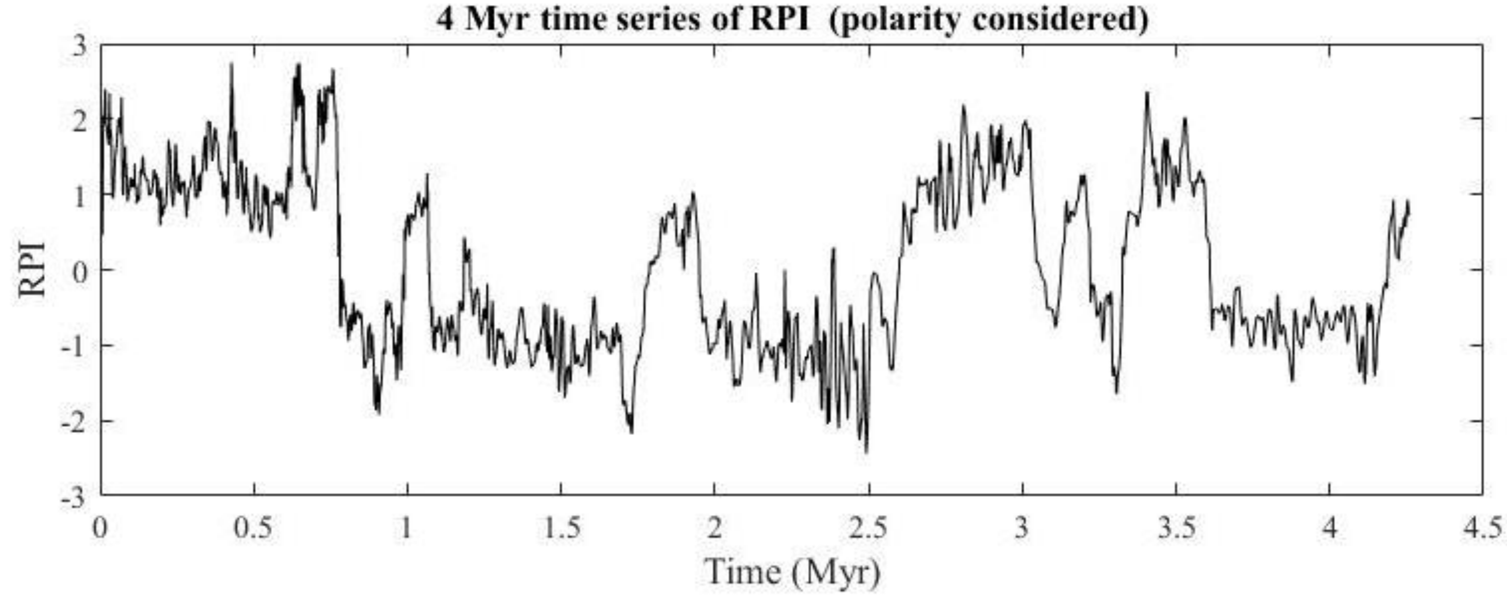


Time series (4 Myr time series)

- 4 Myr long series of palaeomagnetic dipolar moment estimations (4.2566 Myr precisely)
- This series is constructed by analyzing samples drilled from the floor of the Indian Ocean (Meynadier et al., 1994)
- During this time interval, have occurred several inversions of the dipolar geomagnetic field

Time series (4 Myr time series)



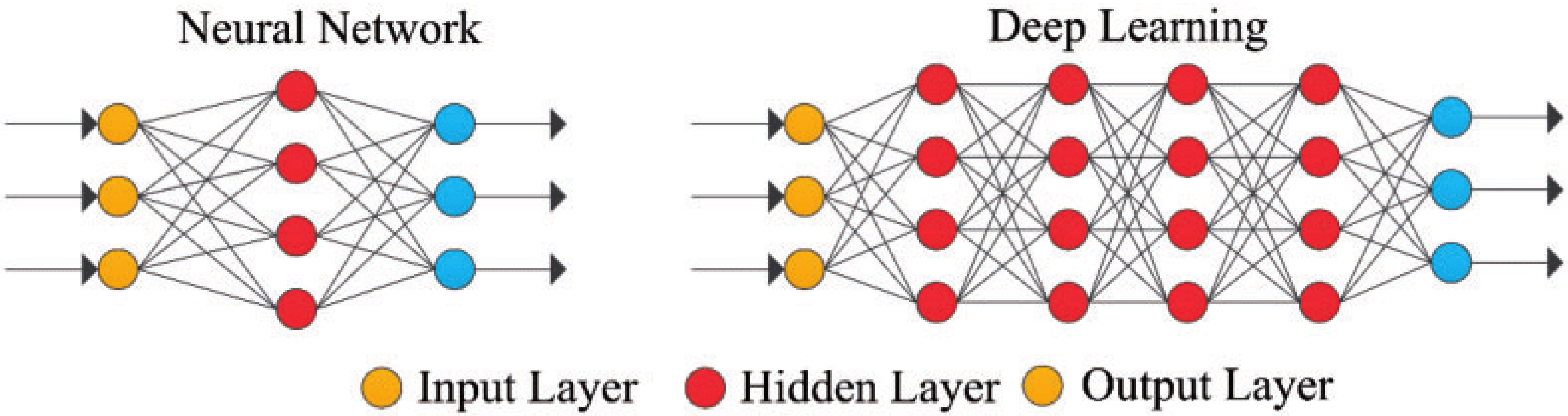


Time series (4 Myr time series)

- 4 Myr long series of palaeomagnetic dipolar moment estimations (4.2566 Myr precisely)
- This series is constructed by analyzing samples drilled from the floor of the Indian Ocean (Meynadier et al., 1994)
- During this time interval, have occurred several inversions of the dipolar geomagnetic field
- The actual series contains relative palaeo-intensity (RPI) data
- They can be converted into absolute palaeo-intensity (API) data through a simple multiplicative gauging that does not affect the statistical properties of the series
- The data are not evenly spaced time-wise and the average timestep is approximately 2 kyr
- The whole series contains 2160 entries

Recursive Neural Network (RNN)

- A new deep learning paradigm is gaining more and more acknowledgment: Recursive Neural Network (RNN) (Chinea, 2009)



- Recursive Neural Networks are non-linear adaptive models that can learn deep structured information
- There is some concern in broadly accepting them:
  - Inherent complexity
  - Computationally expensive learning phase
- Recently more extensively applied in economics (Moghar and Hamiche, 2020), ecological systems (Chen et al., 2018), weather forecasting (Singh et al., 2019), hydrology (Jia et al., 2018)

Motivation

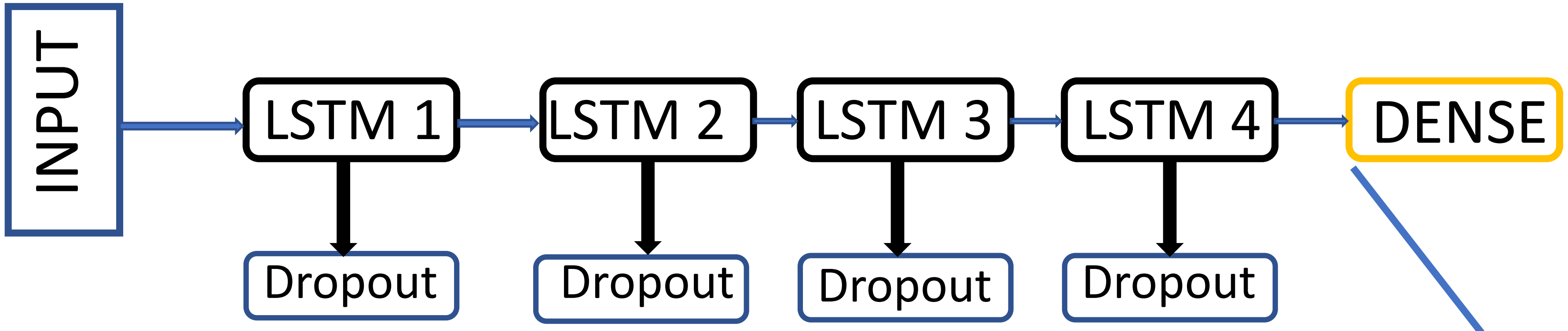
- Construct a Recursive Neural Network (RNN) that can analyze the actual time series
- Analyze
  - Dipolar moment magnitude (positive values)
  - Dipolar moment polarity
- Extend the prediction beyond the actual time series (future objective)

Motivation and procedure

- Construct a Recursive Neural Network (RNN) that can analyze the actual time series
- Analyze
  - Dipolar moment magnitude (positive values)
  - Dipolar moment polarity
- Extend the prediction beyond the actual time series (future objective)
- Train the RNN with a part of the 4 Myr time series
- Afterwards the RNN will provide a prediction that will be compared with the remaining chunk of the original series (validation series)
- Analyze the statistical properties of the predicted and validation series

Building and Training the RNN

- The proposed LSTM architecture



INPUT

LSTM 1

LSTM 2

LSTM 3

LSTM 4

DENSE

Dropout

Dropout

Dropout

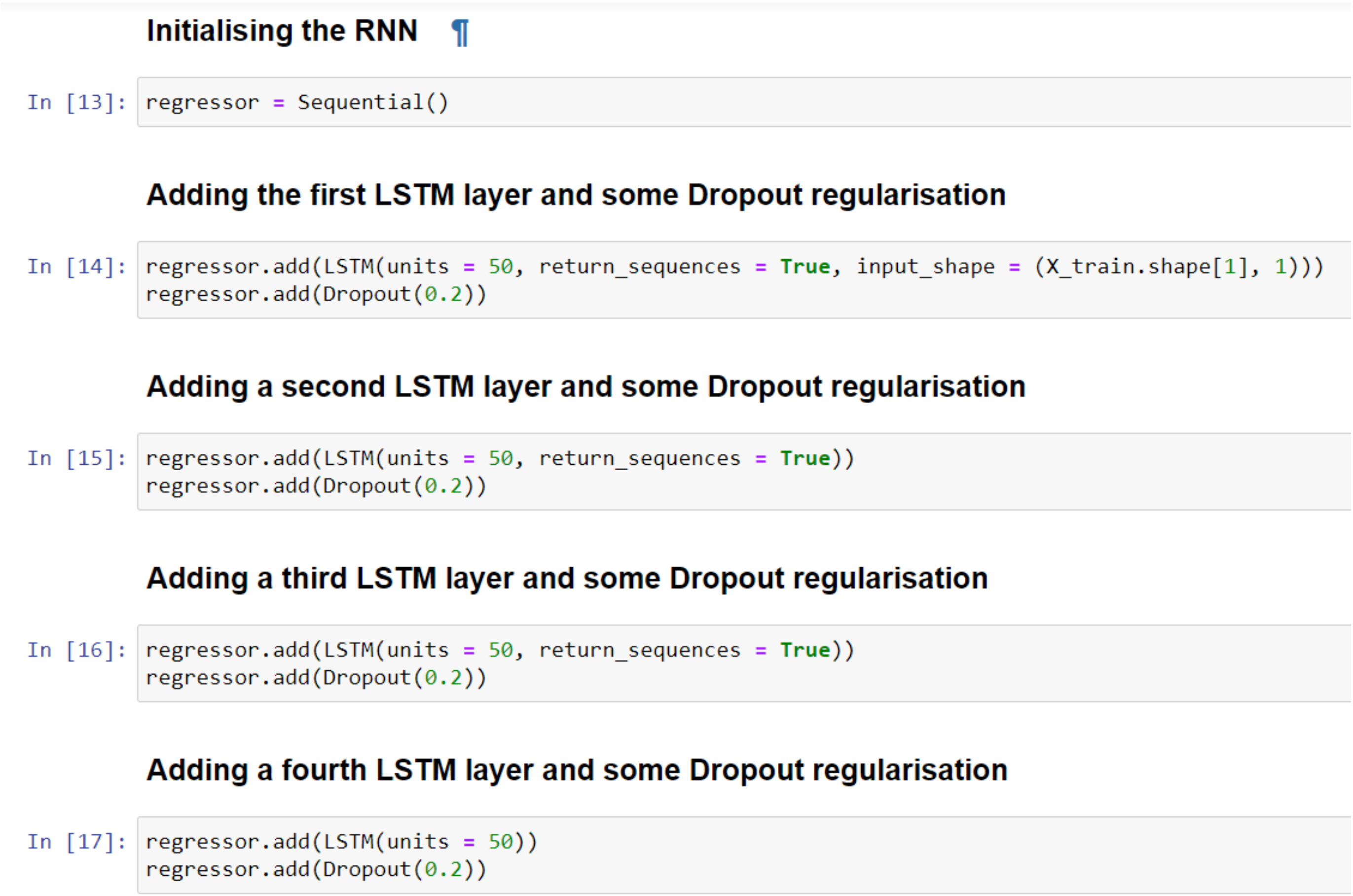
Dropout

- LSTM: Long Short-Term Memory (Learn from sequences of observations and well suited for time series forecasting)
- 4 Layers of LSTM
- Each layer has 50 neurons
- Dropout is 0.2 (to avoid over-fitting!)
- Output layer, DENSE, has only 1 neuron

jhoxha@epoka.edu.al
auka@epoka.edu.al

Building and Training the RNN

- LSTM architecture implementation with Tensorflow and Keras API



Initialising the RNN ¶

```
In [13]: regressor = Sequential()
```

Adding the first LSTM layer and some Dropout regularisation

```
In [14]: regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
         regressor.add(Dropout(0.2))
```

Adding a second LSTM layer and some Dropout regularisation

```
In [15]: regressor.add(LSTM(units = 50, return_sequences = True))
         regressor.add(Dropout(0.2))
```

Adding a third LSTM layer and some Dropout regularisation

```
In [16]: regressor.add(LSTM(units = 50, return_sequences = True))
         regressor.add(Dropout(0.2))
```

Adding a fourth LSTM layer and some Dropout regularisation

```
In [17]: regressor.add(LSTM(units = 50))
         regressor.add(Dropout(0.2))
```

Building and Training the RNN

Adding the output layer

```
In [18]: regressor.add(Dense(units = 1))
```

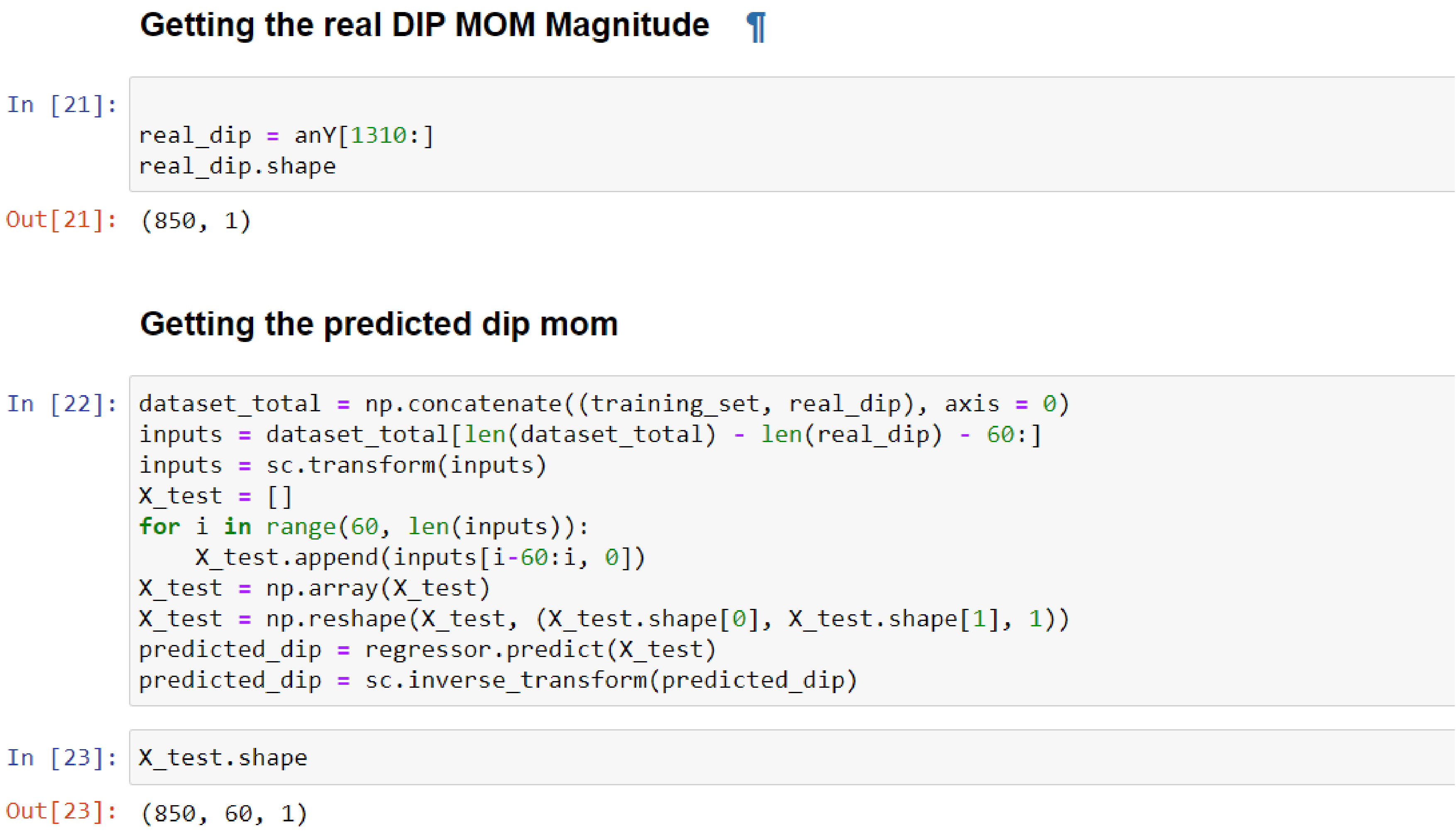
Compiling the RNN

```
In [19]: regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

Fitting the RNN to the Training set

```
In [20]: regressor.fit(X_train, y_train, epochs = 50, batch_size = 32)
```

Testing the RNN



Getting the real DIP MOM Magnitude ¶

```
In [21]:
real_dip = anY[1310:]
real_dip.shape

Out[21]: (850, 1)
```

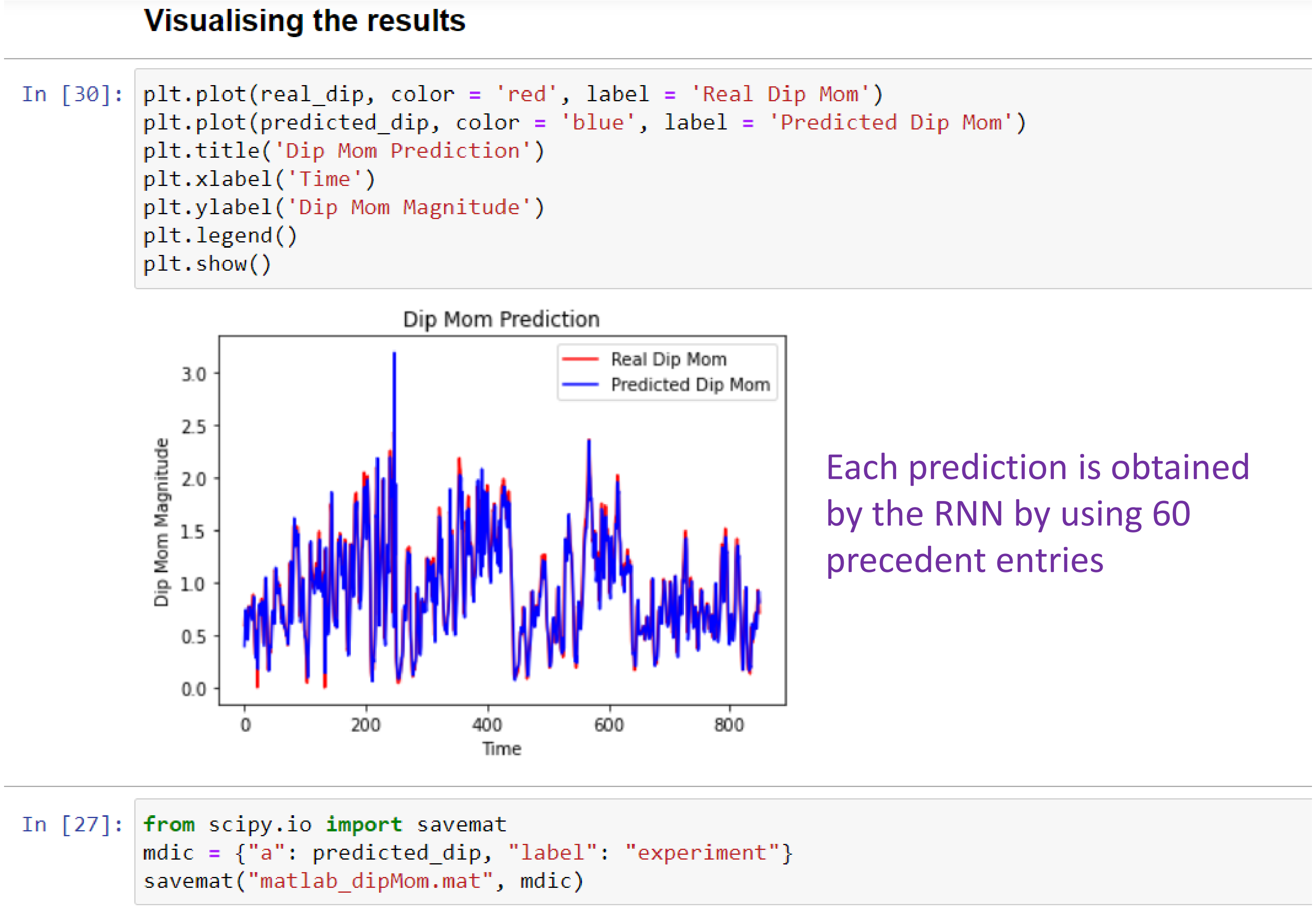
Getting the predicted dip mom

```
In [22]: dataset_total = np.concatenate((training_set, real_dip), axis = 0)
         inputs = dataset_total[len(dataset_total) - len(real_dip) - 60:]
         inputs = sc.transform(inputs)
         X_test = []
         for i in range(60, len(inputs)):
             X_test.append(inputs[i-60:i, 0])
         X_test = np.array(X_test)
         X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
         predicted_dip = regressor.predict(X_test)
         predicted_dip = sc.inverse_transform(predicted_dip)

In [23]: X_test.shape

Out[23]: (850, 60, 1)
```

Testing the RNN



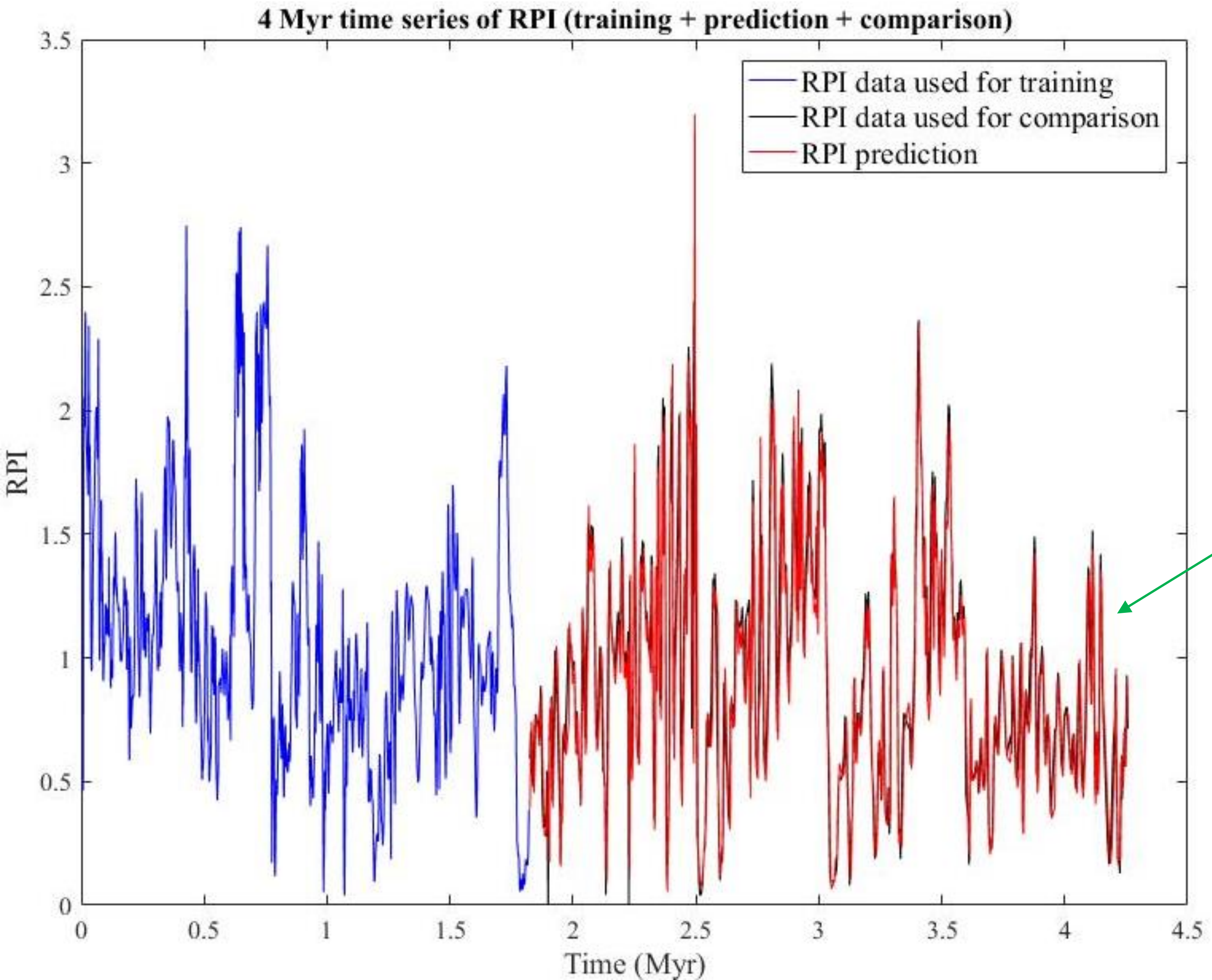
Visualising the results

```
In [30]: plt.plot(real_dip, color = 'red', label = 'Real Dip Mom')
         plt.plot(predicted_dip, color = 'blue', label = 'Predicted Dip Mom')
         plt.title('Dip Mom Prediction')
         plt.xlabel('Time')
         plt.ylabel('Dip Mom Magnitude')
         plt.legend()
         plt.show()
```

```
In [27]: from scipy.io import savemat
         mdic = {"a": predicted_dip, "label": "experiment"}
         savemat("matlab_dipMom.mat", mdic)
```

Each prediction is obtained by the RNN by using 60 precedent entries

Training and Prediction



Last 850 entries

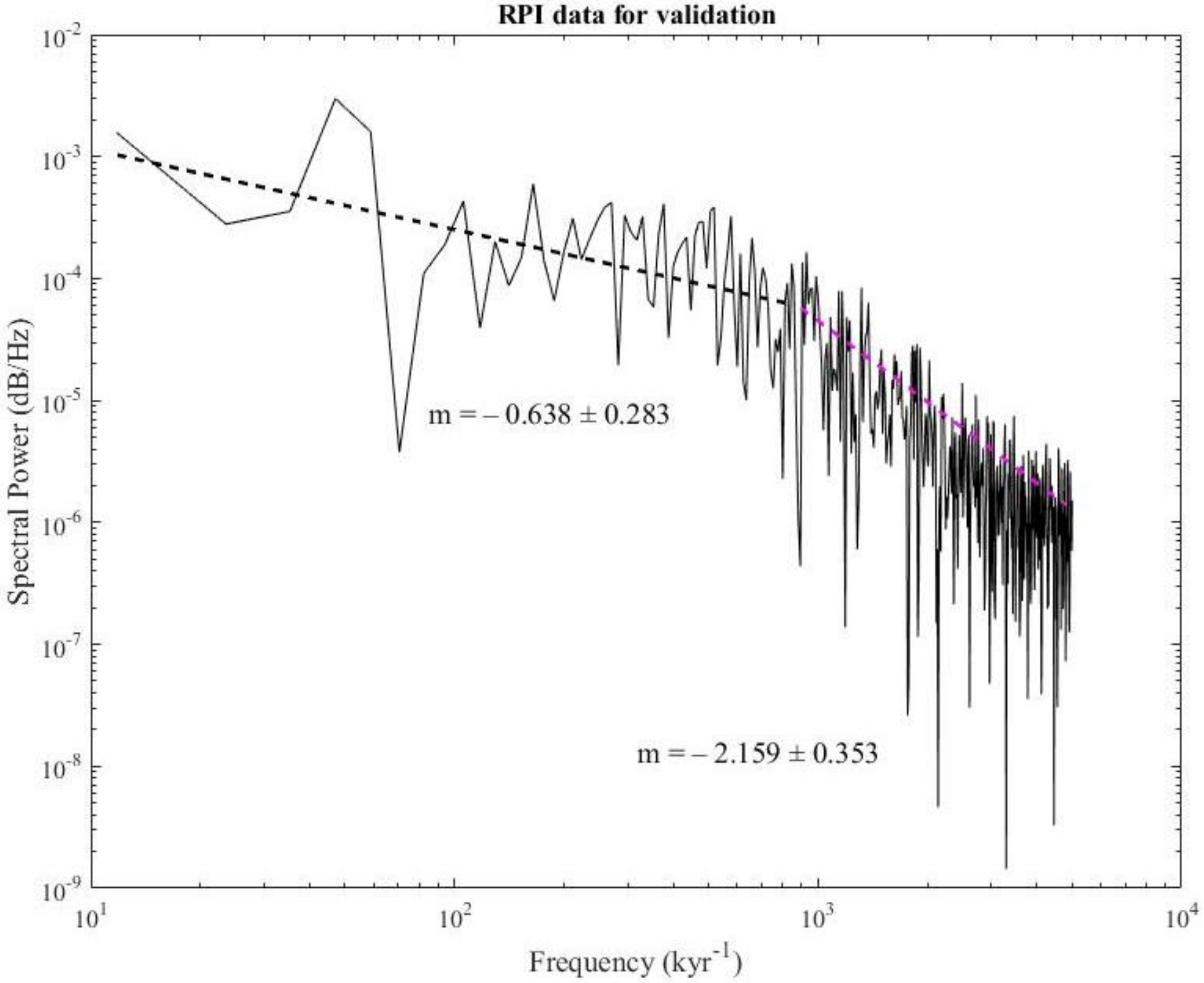
Statistical Properties (to analyze the predicted series)

- The predicted and original time series look very alike. How certain can we be about it?

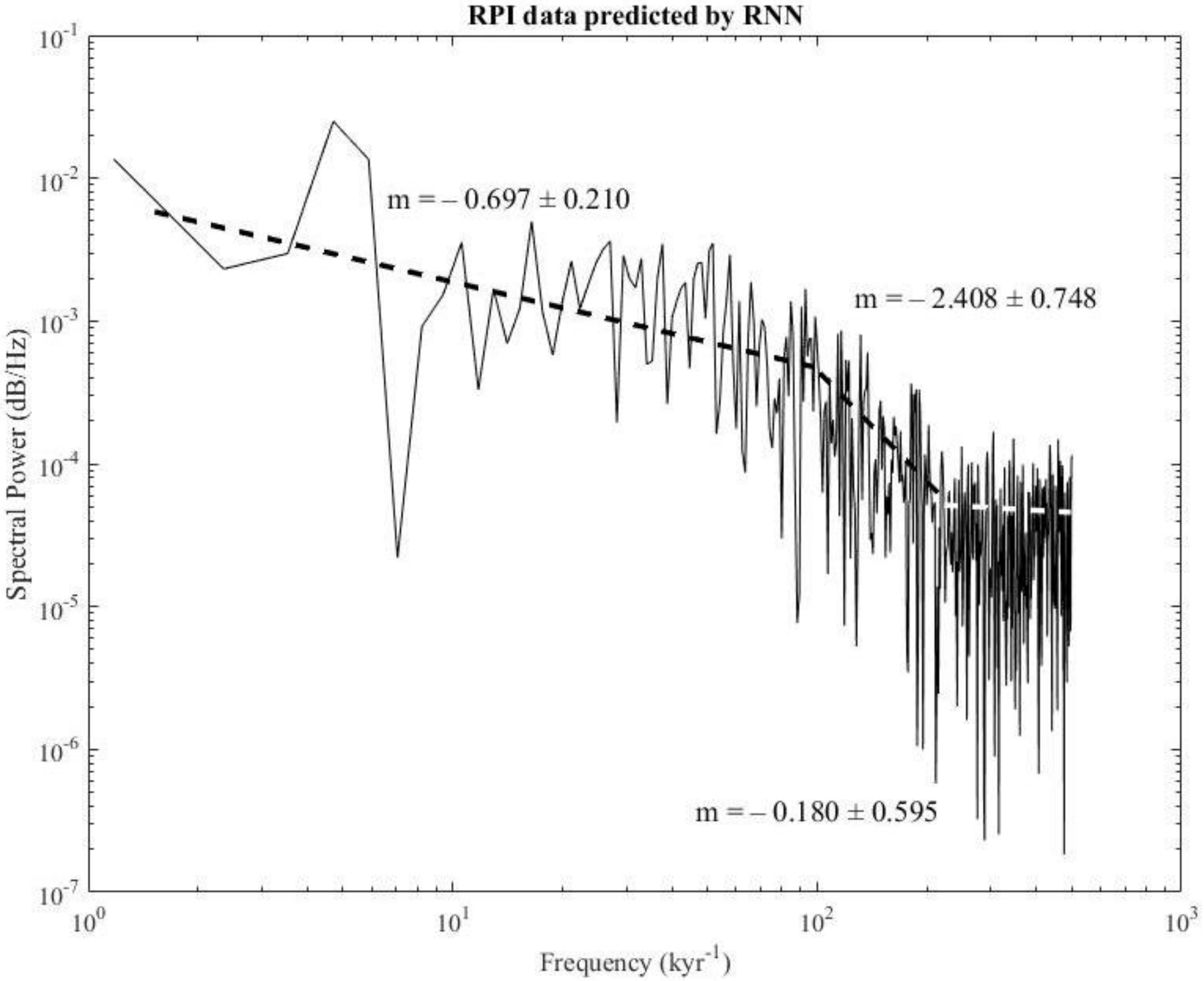
- Power Spectral Density (PSD): distribution of average power against frequency for the predicted and validation time series

- This graph provides valuable information about the statistical properties of e given time series (not the only one)

Power Spectral Density (PSD)

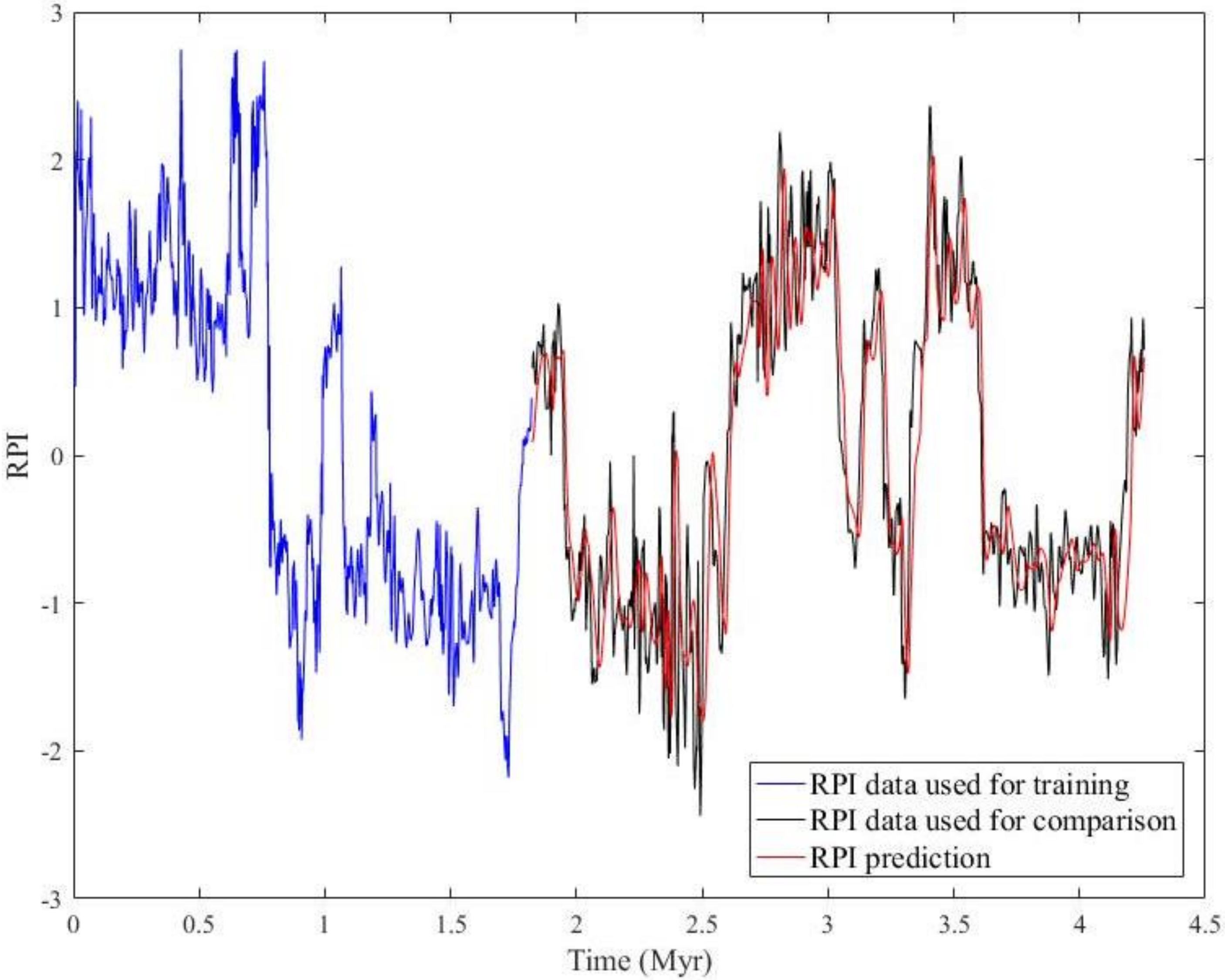


PSD of the validation data series

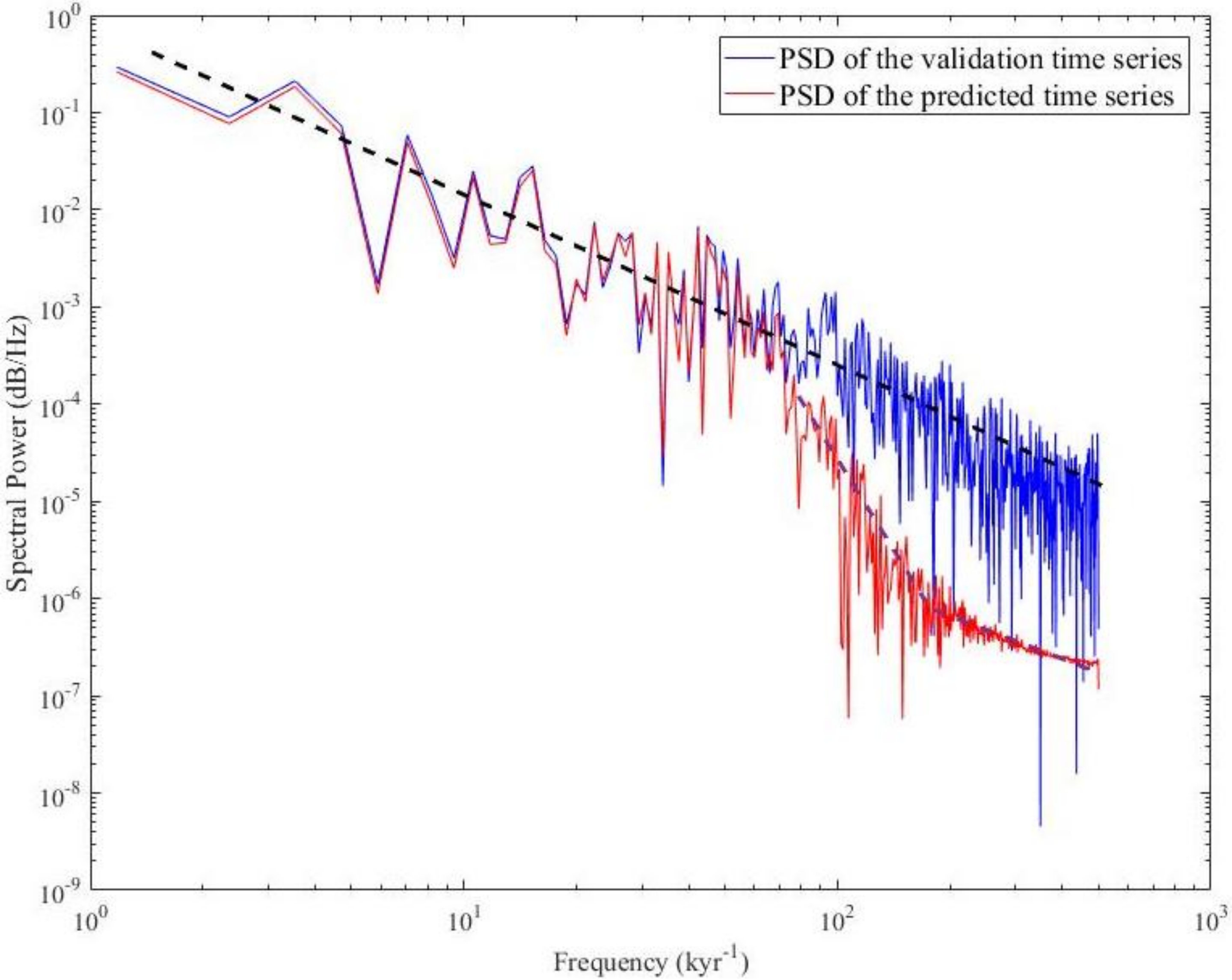


PSD of the predicted series

Training and Prediction (polarity considered)



PSD (polarity considered)



Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right) + \eta \nabla^2 \vec{B}$$

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right) + \cancel{\eta \nabla^2 \vec{B}}$$

(hypothesis “frozen flux”)

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right)$$

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right)$$

- The fluid flow is two-dimensional ($v_r = 0$)

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right)$$

- The fluid flow is two-dimensional ($v_r = 0$)
- There is only one equation, hence there is inherent non-uniqueness

Velocity field at CMB

- The fluid in the outer core and CMB is considered to be and ideal conductor

$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left(\vec{v} \times \vec{B} \right)$$

- The fluid flow is two-dimensional ($v_r = 0$)
- There is only one equation, hence there is inherent non-uniqueness
- Constraints can be applied on the fluid motion (we apply none!)

Velocity field at CMB

- Only the radial component of the magnetic field is continuous through the CMB!
- Radial induction equation $\dot{B}_r = -\nabla \cdot \left(B_r \vec{v} \right)$
- Nabla expanded: $\nabla = \hat{\vec{r}} \left(\hat{\vec{r}} \cdot \nabla \right) + \nabla_H = \frac{\partial}{\partial r} \hat{\vec{r}} + \nabla_H$

$$\nabla_H = \frac{1}{r} \frac{\partial}{\partial \theta} \hat{\vec{\theta}} + \frac{1}{r \sin \theta} \frac{\partial}{\partial \varphi} \hat{\vec{\varphi}}$$

- The divergence yields: $\dot{B}_r + \nabla_H \cdot \left(B_r \vec{v} \right) = 0.$

Velocity field at CMB

- Only the radial component of the magnetic field is continuous through the CMB!
- Radial induction equation $\dot{B}_r = -\nabla\cdot\left(B_r\vec{v}\right)$

- Nabla expanded:

$$\nabla = \hat{\vec{r}}\left(\hat{\vec{r}}\cdot\nabla\right)+\nabla_H = \frac{\partial}{\partial r}\hat{\vec{r}}+\nabla_H$$

$$\nabla_H = \frac{1}{r}\frac{\partial}{\partial\theta}\hat{\vec{\theta}}+\frac{1}{r\sin\theta}\frac{\partial}{\partial\varphi}\hat{\vec{\varphi}}$$

- The divergence yields: $\boxed{\dot{B}_r+\nabla_H\cdot\left(B_r\vec{v}\right)=0.}$

We work with this equation

Velocity field at CMB

- The velocity is separated into a toroidal and poloidal constituent (Backus, 1986):

$$\vec{v}_T = \nabla \times \left(\vec{r} T \right) = \left(0, \frac{1}{\sin\theta} \frac{\partial T}{\partial \varphi}, -\frac{\partial T}{\partial \theta} \right),$$

$$\vec{v}_S = \nabla \times \left[\nabla \times \left(\vec{r} S \right) \right] = \nabla_H \left(rS \right) = \left(0, \frac{\partial S}{\partial \theta}, \frac{1}{\sin\theta} \frac{\partial S}{\partial \varphi} \right)$$

- Total velocity $\vec{v} = \left(0, \frac{1}{\sin\theta} \frac{\partial T}{\partial \varphi} + \frac{\partial S}{\partial \theta}, -\frac{\partial T}{\partial \theta} + \frac{1}{\sin\theta} \frac{\partial S}{\partial \varphi} \right)$
- After substitution

$$\dot{B}_r = -B_r \left(\frac{1}{r} \frac{\cos\theta}{\sin\theta} \frac{\partial S}{\partial \theta} + \frac{1}{r} \frac{\partial^2 S}{\partial \theta^2} + \frac{1}{r \sin^2\theta} \frac{\partial^2 S}{\partial \varphi^2} \right)$$

$$-\left(\frac{1}{r \sin\theta} \frac{\partial T}{\partial \varphi} + \frac{1}{r} \frac{\partial S}{\partial \theta} \right) \frac{\partial B_r}{\partial \theta} + \left(\frac{1}{r \sin\theta} \frac{\partial T}{\partial \theta} - \frac{1}{r \sin^2\theta} \frac{\partial S}{\partial \varphi} \right) \frac{\partial B_r}{\partial \varphi}$$

Velocity field at CMB

- The velocity is separated into a toroidal and poloidal constituent (Backus, 1986)

$$\vec{v}_T = \nabla\times\left(\vec{r}T\right) = \left(0, \frac{1}{\sin\theta}\frac{\partial T}{\partial\varphi}, -\frac{\partial T}{\partial\theta}\right),$$

$$\vec{v}_S = \nabla\times\left[\nabla\times\left(\vec{r}S\right)\right] = \nabla_H\left(rS\right) = \left(0, \frac{\partial S}{\partial\theta}, \frac{1}{\sin\theta}\frac{\partial S}{\partial\varphi}\right)$$

- Total velocity $\vec{v} = \left(0, \frac{1}{\sin\theta}\frac{\partial T}{\partial\varphi} + \frac{\partial S}{\partial\theta}, -\frac{\partial T}{\partial\theta} + \frac{1}{\sin\theta}\frac{\partial S}{\partial\varphi}\right)$
- After substitution

$$\dot{B}_r = -B_r\left(\frac{1}{r}\frac{\cos\theta}{\sin\theta}\frac{\partial S}{\partial\theta} + \frac{1}{r}\frac{\partial^2 S}{\partial\theta^2} + \frac{1}{r\sin^2\theta}\frac{\partial^2 S}{\partial\varphi^2}\right)$$

$$-\left(\frac{1}{r\sin\theta}\frac{\partial T}{\partial\varphi} + \frac{1}{r}\frac{\partial S}{\partial\theta}\right)\frac{\partial B_r}{\partial\theta} + \left(\frac{1}{r\sin\theta}\frac{\partial T}{\partial\theta} - \frac{1}{r\sin^2\theta}\frac{\partial S}{\partial\varphi}\right)\frac{\partial B_r}{\partial\varphi}$$

Velocity field at CMB

- In spherical harmonics with complex coefficients

$$B_r = \sum_{l_1, m_1} \left(\frac{a}{r} \right)^{l_1+2} \left(l_1 + 1 \right) g_{l_1}^{m_1} Y_{l_1}^{m_1} \left(\theta, \varphi \right)$$

$$T = \sum_{l_2, m_2} t_{l_2}^{m_2} Y_{l_2}^{m_2} \left(\theta, \varphi \right)$$

$$S = \sum_{l_3, m_3} s_{l_3}^{m_3} Y_{l_3}^{m_3} \left(\theta, \varphi \right)$$

$$\dot{B}_r = \frac{\partial B_r}{\partial t} = \sum_{l_1, m_1} \left(\frac{a}{r} \right)^{l_1+2} \left(l_1 + 1 \right) \dot{g}_{l_1}^{m_1} Y_{l_1}^{m_1} \left(\theta, \varphi \right)$$

- The coefficients *t* and *s* are unknown

Velocity field at CMB

- Substitution into the radial induction equation

$$\sum_{l_1,m_1}\left(\frac{a}{r}\right)^{l_1+2}\left(l_1+1\right)\dot{g}_{l_1}^{m_1}Y_{l_1}^{m_1}\left(\theta,\varphi\right)=\frac{1}{r}\sum_{l_2,m_2}\sum_{l_3,m_3}\left(\frac{a}{r}\right)^{l_2+2}\left(l_2+1\right)g_{l_2}^{m_2}\times$$

$$\times[\![t_{l_3}^{m_3}\frac{1}{\sin\theta}\left[\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}-\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right]-$$

$$-s_{l_3}^{m_3}\left[\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}+\frac{1}{\sin^2\theta}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}-l_3\left(l_3+1\right)Y_{l_2}^{m_2}Y_{l_3}^{m_3}\right]]\!].$$

- Integration over the CMB (Whaler, 1986)

$$\dot{g}_{l_1}^{m_1}=\sum_{l_3,m_3}\left[\frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{\left(l_1+1\right)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}\left(l_2+1\right)g_{l_2}^{m_2}\times\oint_{\Omega}\frac{1}{\sin\theta}Y_{l_1}^{m_1*}\left(\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}-\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right)d\Omega\right]t_{l_3}^{m_3}+$$

$$+\sum_{l_3,m_3}\left\{-\frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{\left(l_1+1\right)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}\left(l_2+1\right)g_{l_2}^{m_2}\times\oint_{\Omega}\left[Y_{l_1}^{m_1*}\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}+\frac{1}{\sin^2\theta}Y_{l_1}^{m_1*}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}-l_3\left(l_3+1\right)Y_{l_1}^{m_1*}Y_{l_2}^{m_2}Y_{l_3}^{m_3}\right]d\Omega\right\}s_{l_3}^{m_3}$$

Velocity field at CMB

- Substitution into the radial induction equation

$$\sum_{l_1,m_1}\left(\frac{a}{r}\right)^{l_1+2}(l_1+1)\dot{g}_{l_1}^{m_1}Y_{l_1}^{m_1}(\theta,\varphi)=\frac{1}{r}\sum_{l_2,m_2}\sum_{l_3,m_3}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)g_{l_2}^{m_2}\times$$

$$\times\Bigg[\!\!\Bigg[t_{l_3}^{m_3}\frac{1}{\sin\theta}\left[\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}-\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right]-$$

$$-s_{l_3}^{m_3}\left[\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}+\frac{1}{\sin^2\theta}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}-l_3(l_3+1)Y_{l_2}^{m_2}Y_{l_3}^{m_3}\right]\Bigg]\!\!\Bigg].$$

SV of the magnetic field

Magnetic field

- Integration over the CMB

$$\dot{g}_{l_1}^{m_1}=\sum_{l_3,m_3}\left[\frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)g_{l_2}^{m_2}\times\oint_{\Omega}\frac{1}{\sin\theta}Y_{l_1}^{m_1*}\left(\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}-\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right)d\Omega\right]\boxed{t_{l_3}^{m_3}}+$$

$$+\sum_{l_3,m_3}\left\{-\frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)g_{l_2}^{m_2}\times\oint_{\Omega}\left[Y_{l_1}^{m_1*}\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}+\frac{1}{\sin^2\theta}Y_{l_1}^{m_1*}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}-l_3(l_3+1)Y_{l_1}^{m_1*}Y_{l_2}^{m_2}Y_{l_3}^{m_3}\right]d\Omega\right\}\boxed{s_{l_3}^{m_3}}$$

Toroidal and Poloidal velocity fields

Velocity field at CMB

- Define the Elsasser and Gaunt matrices

$$E_{l_1 l_3}^{m_1 m_3} = \frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2} \frac{1}{\left(l_1+1\right)} \sum_{l_2, m_2} \left(\frac{a}{r}\right)^{l_2+2} \left(l_2+1\right) g_{l_2}^{m_2} \times \oint_{\Omega} \left(\frac{\partial Y_{l_3}^{m_3}}{\partial \theta} \frac{\partial Y_{l_2}^{m_2}}{\partial \varphi} - \frac{\partial Y_{l_2}^{m_2}}{\partial \theta} \frac{\partial Y_{l_3}^{m_3}}{\partial \varphi} \right) Y_{l_1}^{m_1} d\Omega$$

$$G_{l_1 l_3}^{m_1 m_3} = \frac{2}{r}\left(\frac{r}{a}\right)^{l_1+2} \frac{1}{\left(l_1+1\right)} \sum_{l_2, m_2} \left(\frac{a}{r}\right)^{l_2+2} \left(l_2+1\right) g_{l_2}^{m_2} \times \left[l_1\left(l_1+1\right) + l_3\left(l_3+1\right) - l_2\left(l_2+1\right) \right] \oint_{\Omega} Y_{l_1}^{m_1} Y_{l_2}^{m_2} Y_{l_3}^{m_3} d\Omega$$

Velocity field at CMB

- Define the Elsasser and Gaunt matrices

Elsasser integral

$$E_{l_1 l_3}^{m_1 m_3} = \frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2} \frac{1}{(l_1+1)} \sum_{l_2,m_2} \left(\frac{a}{r}\right)^{l_2+2} (l_2+1)\, g_{l_2}^{m_2} \times \oint_{\Omega} \left(\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi} - \frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right) Y_{l_1}^{m_1}\, d\Omega$$

Gaunt integral

$$G_{l_1 l_3}^{m_1 m_3} = \frac{2}{r}\left(\frac{r}{a}\right)^{l_1+2} \frac{1}{(l_1+1)} \sum_{l_2,m_2} \left(\frac{a}{r}\right)^{l_2+2} (l_2+1)\, g_{l_2}^{m_2} \times \left[l_1(l_1+1)+l_3(l_3+1)-l_2(l_2+1)\right] \oint_{\Omega} Y_{l_1}^{m_1} Y_{l_2}^{m_2} Y_{l_3}^{m_3}\, d\Omega$$

Velocity field at CMB

- Define the Elsasser and Gaunt matrices

Elsasser
integral

$$E^{m_1 m_3}_{l_1 l_3} = \frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)\,g^{m_2}_{l_2}\times\oint_{\Omega}\left(\frac{\partial Y^{m_3}_{l_3}}{\partial\theta}\frac{\partial Y^{m_2}_{l_2}}{\partial\varphi}-\frac{\partial Y^{m_2}_{l_2}}{\partial\theta}\frac{\partial Y^{m_3}_{l_3}}{\partial\varphi}\right)Y^{m_1}_{l_1}\,d\Omega$$

Gaunt
integral

$$G^{m_1 m_3}_{l_1 l_3} = \frac{2}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)\,g^{m_2}_{l_2}\times\left[l_1(l_1+1)+l_3(l_3+1)-l_2(l_2+1)\right]\oint_{\Omega}Y^{m_1}_{l_1}Y^{m_2}_{l_2}Y^{m_3}_{l_3}\,d\Omega$$

- In matrix form: $\dot{\mathbf{g}} = \mathbf{Et} + \mathbf{Gs}$

Velocity field at CMB

- Define the Elsasser and Gaunt matrices

Elsasser integral

$$E_{l_1 l_3}^{m_1 m_3} = \frac{1}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)\,g_{l_2}^{m_2}\times\oint_\Omega\left(\frac{\partial Y_{l_3}^{m_3}}{\partial\theta}\frac{\partial Y_{l_2}^{m_2}}{\partial\varphi}-\frac{\partial Y_{l_2}^{m_2}}{\partial\theta}\frac{\partial Y_{l_3}^{m_3}}{\partial\varphi}\right)Y_{l_1}^{m_1}\,d\Omega$$

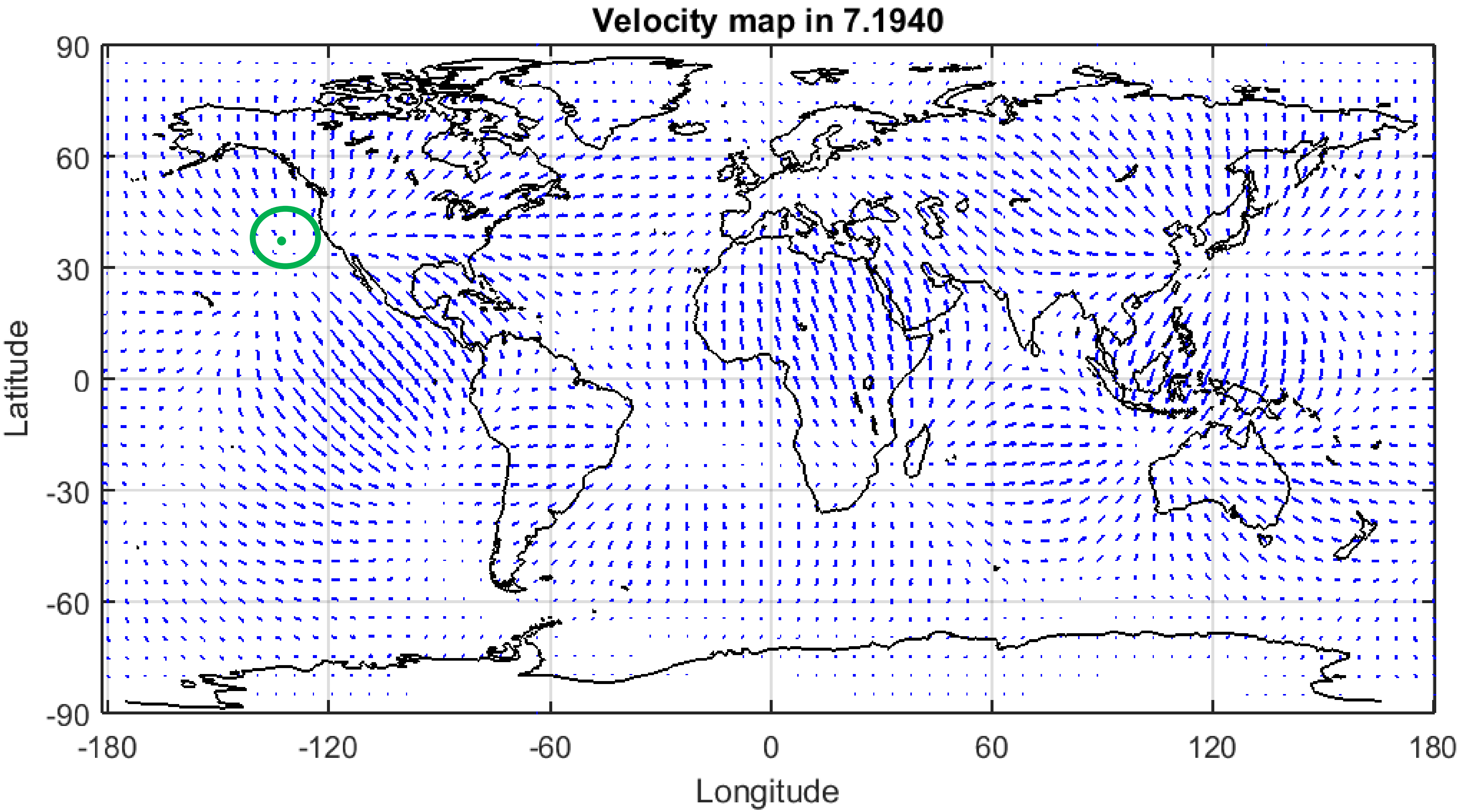
Gaunt integral

$$G_{l_1 l_3}^{m_1 m_3} = \frac{2}{r}\left(\frac{r}{a}\right)^{l_1+2}\frac{1}{(l_1+1)}\sum_{l_2,m_2}\left(\frac{a}{r}\right)^{l_2+2}(l_2+1)\,g_{l_2}^{m_2}\times\left[l_1(l_1+1)+l_3(l_3+1)-l_2(l_2+1)\right]\oint_\Omega Y_{l_1}^{m_1}Y_{l_2}^{m_2}Y_{l_3}^{m_3}\,d\Omega$$

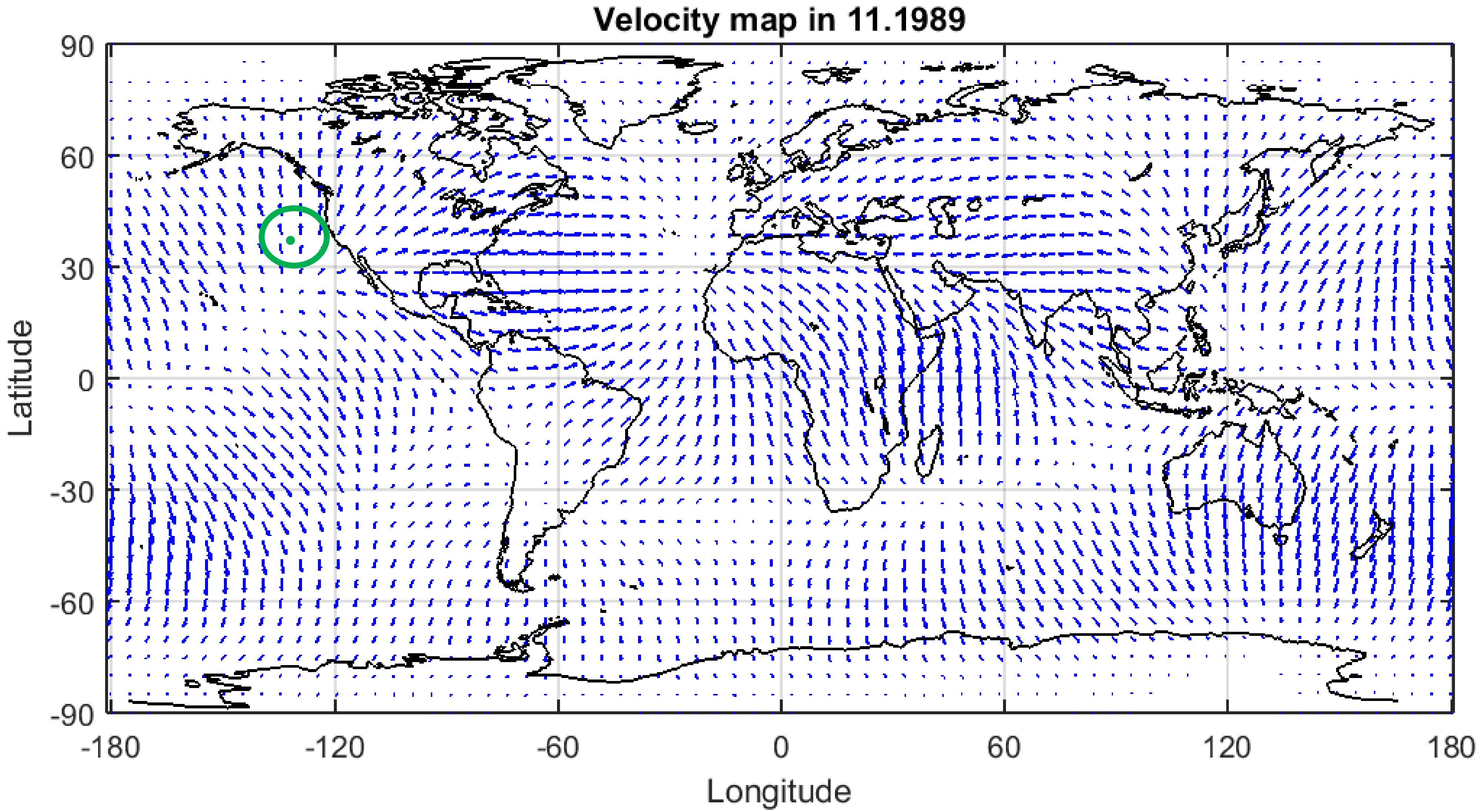
- In matrix form: $\dot{\mathbf{g}} = \mathbf{E}\mathbf{t} + \mathbf{G}\mathbf{s}$

- Final version: $\dot{\mathbf{g}} = (\mathbf{E} : \mathbf{G})\left(\frac{\mathbf{t}}{\mathbf{s}}\right)$

Velocity field at the CMB



Velocity field at the CMB

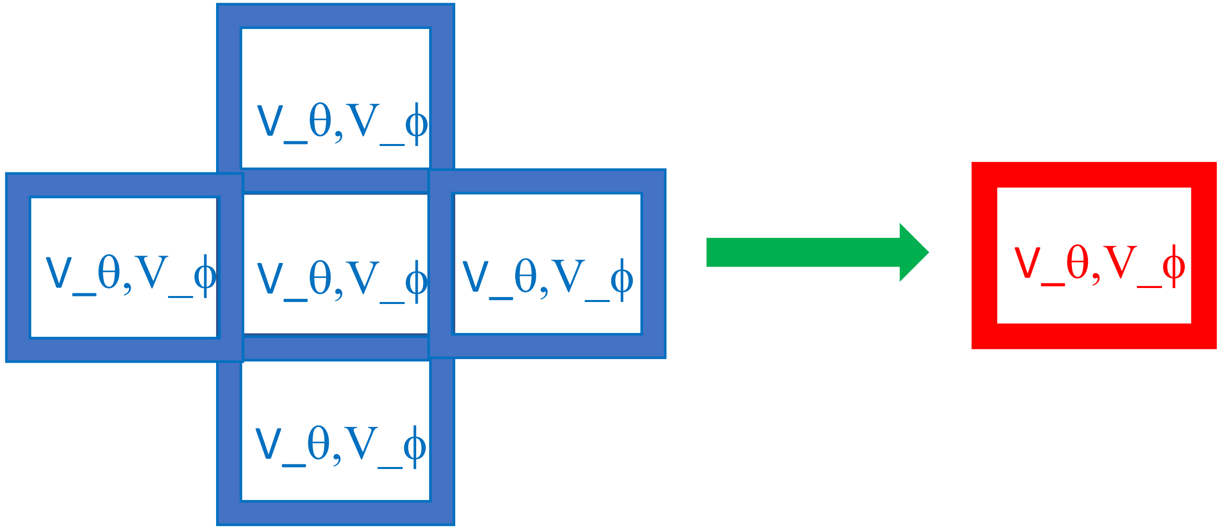


Further specifications

- Gated Recurrent Units (GRU) is like Long-Short Term Memory (LSTM), but with fewer parameters.
- GRU reported to perform better than LSTM for small datasets (Chung et al. 2014; arXiv:1412.3555)
- Since the dataset used (1935-1989) is small, we chose to use GRU
- The effect of the size is ‘indirectly’ observed in the dependency of performance of the neural network on train-test split value [0.75-0.9]
- A split rate smaller than 90% decreases the performance considerably
- The data set is normalized (largest speed is 1)

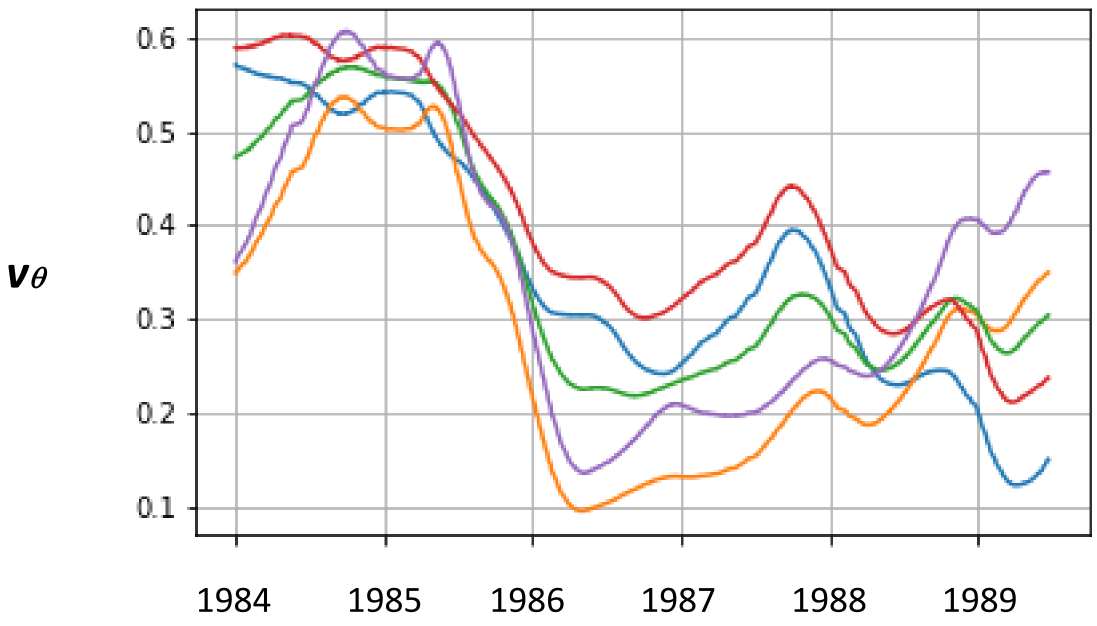
Prediction using RNN

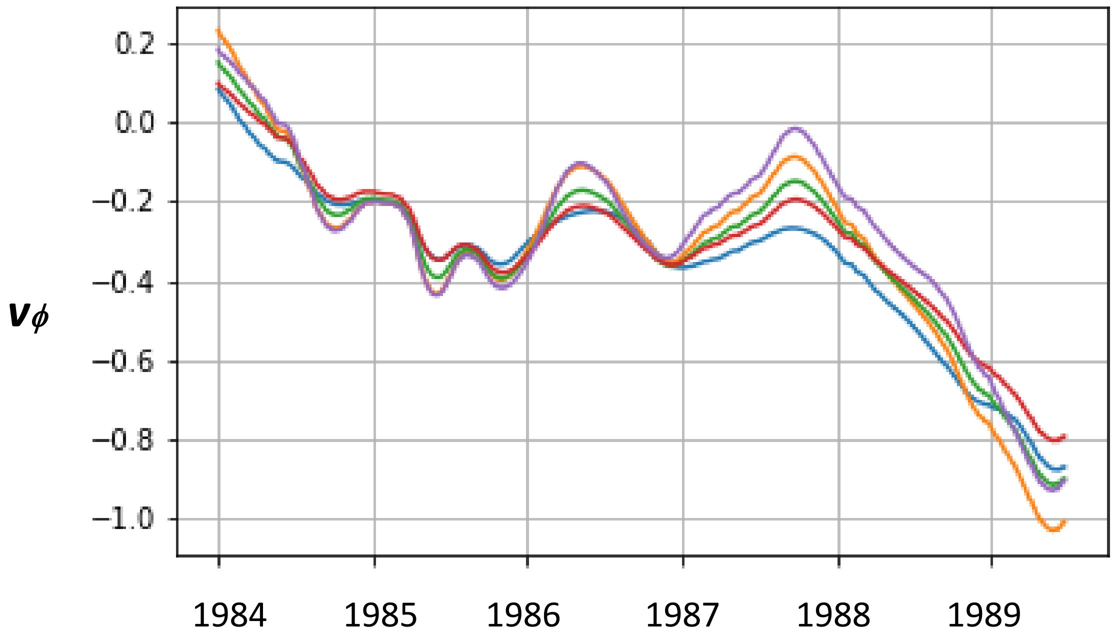
- We use the values of the velocities of 5 regions (1 centrally located at (40 N, 130 W) and 4 neighbouring sites)
- We used data from 1935 until 1990
  - Higher reliability in the latter years
- Recurrent neural network is used to predict “2 vectors” (sequences) after learning from 10 input vectors
- Used: Tensorflow, Keras



Time dependence of velocity components in the selected domain

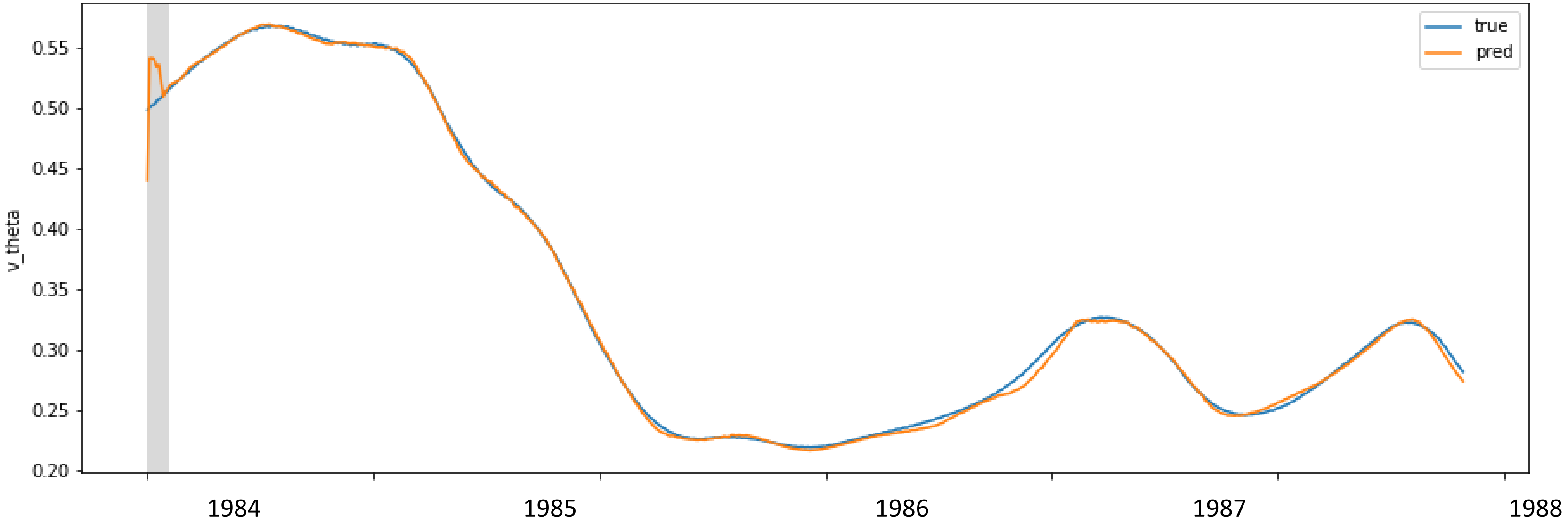
We observe a variance of the 2 components of the velocities for 5 neighbouring sites (Training series spanning the last 10 years)



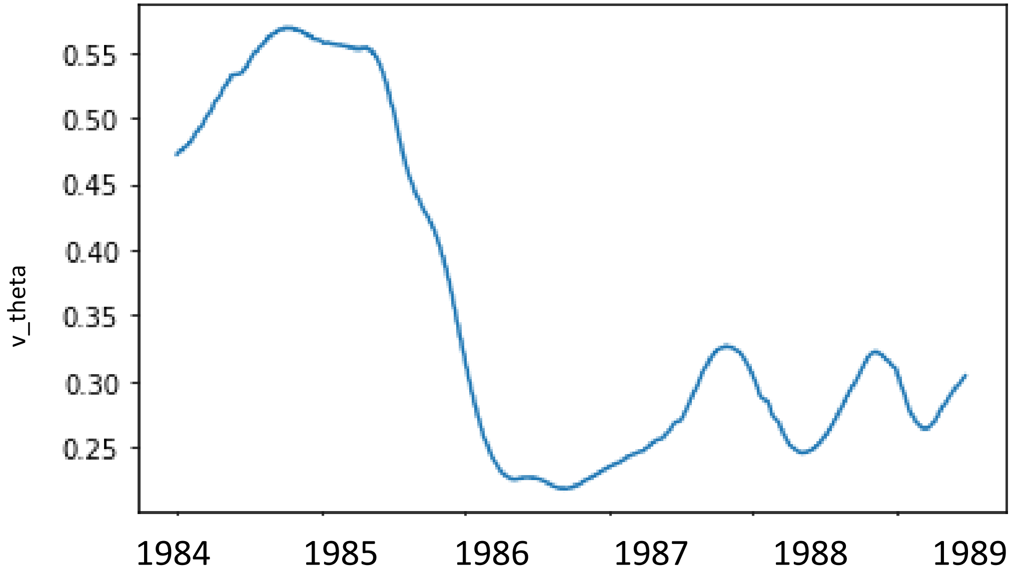


The variations on the point in the map are shown in solid green line

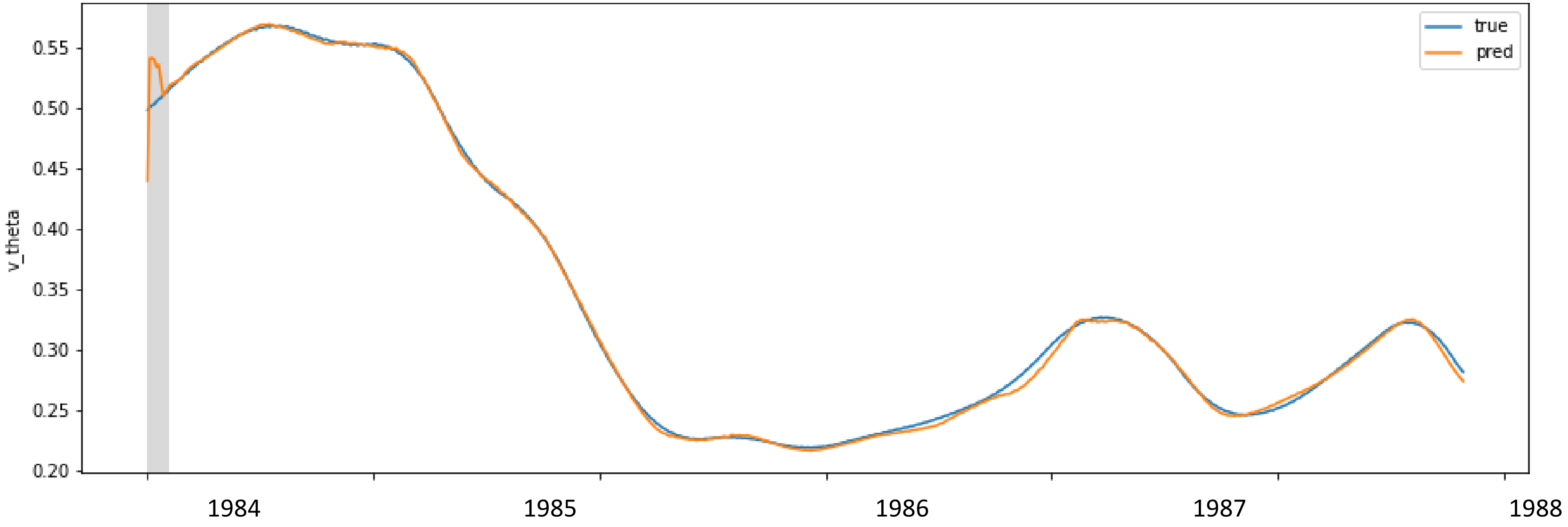
Predictions for the central point (temporal approach)

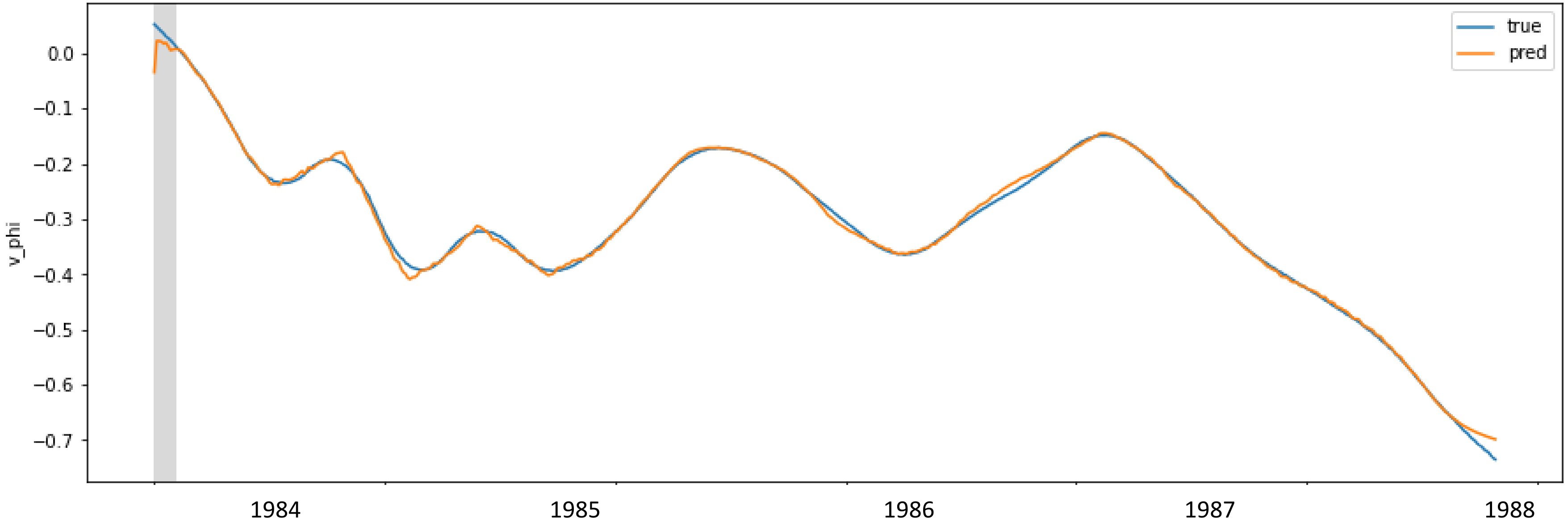


This approach focuses on the time series constructed for each component in a defined observation point. Each training series contains 60–120 entries.

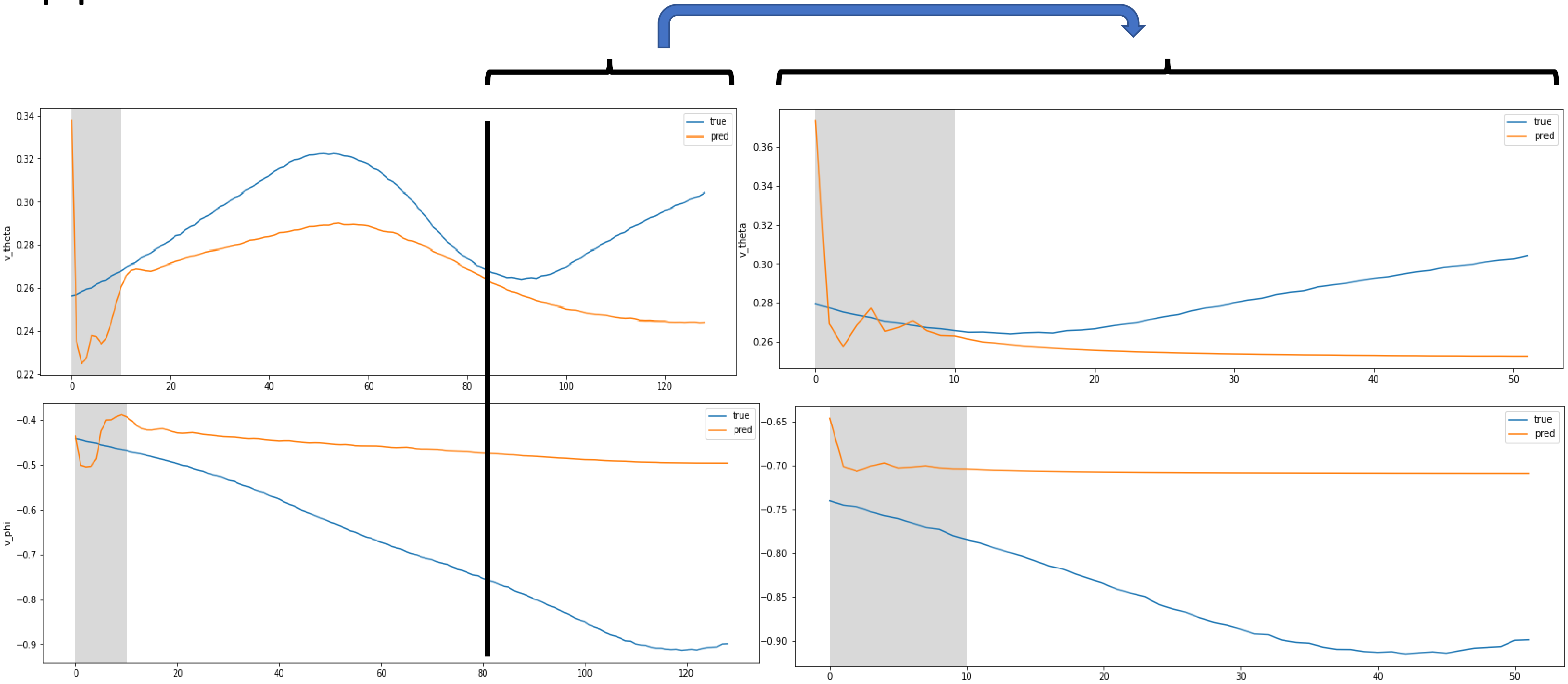


Predictions for the central point (temporal approach)





Predicting future values of v_θ & v_ϕ for the spatial approach



Train 80%

We consider 10 input vectors altogether!

Train 92%

Conclusions

- We observed that the time approach offers a very reliable method on how to possibly predict the velocity field components for a given point at CMB
- We observed a good performance in determining two components of the velocity using RNN based on previous values of the region and 4 neighbouring sites
- (Future work) Will use larger neighbouring regions around a site of interest
- (Future work) Will use a larger dataset (initial year < 1930)
- (Future work) We will extend the dataset with "predictions" beyond 1990
- (Future work) Will construct a model that provides a more realistic velocity field at CMB

Conclusions

- We observed that the time approach offers a very reliable method on how to possibly predict the velocity field components for a given point at CMB
- We observed a good performance in determining two components of the velocity using RNN based on previous values of the region and 4 neighbouring sites
- (Future work) Will use larger neighbouring regions around a site of interest
- (Future work) Will use a larger dataset (initial year < 1930)
- (Future work) We will extend the dataset with “predictions” beyond 1990
- (Future work) Will construct a model that provides a more realistic velocity field at CMB

References

- Meynadier et al., 1994 https://doi.org/10.1016/0012-821X(94)90245-3
- Jan W. Kantelhardt, Stephan A. Zschiegner, Eva Koscielny-Bunde, Shlomo Havlin, Armin Bunde, H.Eugene Stanley. Multifractal detrended fluctuation analysis of nonstationary time series, Physica A: Statistical Mechanics and its Applications, Volume 316, Issues 1–4, 2002, https://doi.org/10.1016/S0378-4371(02)01383-3
- Chinea A. (2009) Understanding the Principles of Recursive Neural Networks: A Generative Approach to Tackle Model Complexity. In: Alippi C., Polycarpou M., Panayiotou C., Ellinas G. (eds) Artificial Neural Networks – ICANN 2009. ICANN 2009. Lecture Notes in Computer Science, vol 5768. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04274-4_98
- Adil Moghar, Mhamed Hamiche. Stock Market Prediction Using LSTM Recurrent Neural Network, Procedia Computer Science, Volume 170, 2020. https://doi.org/10.1016/j.procs.2020.03.049
- Singh, Siddharth and Kaushik, Mayank and Gupta, Ambuj and Malviya, Anil Kumar, Weather Forecasting Using Machine Learning Techniques (March 11, 2019). Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Available at SSRN: https://ssrn.com/abstract=3350281 or http://dx.doi.org/10.2139/ssrn.3350281
- Yingyi Chen, Qianqian Cheng, Yanjun Cheng, Hao Yang, Huihui Yu Applications of Recurrent Neural Networks in Environmental Factor Forecasting: A Review. *Neural Computation* (2018) 30 (11): 2855–2881. https://doi.org/10.1162/neco_a_01134
- Jia et al., 2018. Physics Guided Recurrent Neural Networks For Modeling Dynamical Systems: Application to Monitoring Water Temperature And Quality In Lakes. arXiv:1810.02880v1 **[cs.LG]**

References

- Backus, G., 1986. Poloidal and Toroidal Fields in Geomagnetic Field Modeling. Reviews Of Geophysics, Vol. 24, No. 1, 75-109.
- Whaler, 1986. Geomagnetic evidence for fluid upwelling at the coremantle boundary. Geophys. J. R. astr. SOC. 86, 563-588
- Jackson, A., Jonkers, A. R. T., and Walker, M. R., 2000. Four centuries of geomagnetic secular variation from historical records. Phil. Trans. R. Soc. Lond. 358, 957-990.
- Campuzano SA, Pavón-Carrasco FJ,De Santis A, González-López A and Qamili E (2021) South Atlantic Anomaly Areal Extent as a Possible Indicator of Geomagnetic Jerks in the Satellite Era. Front. Earth Sci. 8:607049. doi: 10.3389/feart.2020.607049
- Chulliat, A., and Maus, S. (2014). Geomagnetic secular acceleration, jerks, and a localized standing wave at the core surface from 2000 to 2010. J. Geophys. Res. Solid Earth. 119, 1531–1543. doi:10.1002/2013JB010604
- Chung et al. 2014; arXiv:1412.3555