NOTES ON PAPERS

Contents

[Original papers given 2](#_Toc112057785)

[Discovering governing equations from data by sparse identification of nonlinear dynamical systems 2](#_Toc112057786)

[SINDy-PI: A Robust Algorithm for Parallel Implicit Sparse Identification of Nonlinear Dynamics 2](#_Toc112057787)

[Physics- informed machine learning 4](#_Toc112057788)

[Simulation and Analysis of Vibration of Rolling Bearing 5](#_Toc112057789)

[Other SINDy variations 6](#_Toc112057790)

[Sparse identification of nonlinear dynamics for rapid model recovery 6](#_Toc112057791)

[SINDy with Control: A Tutorial 7](#_Toc112057792)

[AIC and BIC 8](#_Toc112057793)

[SINDy-SA: Enhancing nonlinear system identification with sensitivity analysis 9](#_Toc112057794)

[Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control 10](#_Toc112057795)

[Other papers related to rolling bearings 11](#_Toc112057796)

[A Deep Learning-based Approach for Fault Diagnosis of Roller Element Bearings 11](#_Toc112057797)

[Nonlinear model and simulation of a rolling bearing 12](#_Toc112057798)

[Effect of the raceway defects on the nonlinear dynamic behavior of rolling 12](#_Toc112057799)

[Other PIML methods 12](#_Toc112057800)

[Discovering unmodeled components in astrodynamics with symbolic regression 12](#_Toc112057801)

[Space applications 12](#_Toc112057802)

[Incorporating Physical Knowledge Into Machine Learning for Planetary Space Physics 12](#_Toc112057803)

[Science Autonomy and Space Science: Application to the ExoMars Mission 13](#_Toc112057804)

[Machine learning in space: extending our reach 13](#_Toc112057805)

[c 14](#_Toc112057806)

[Identifying Exoplanets with Deep Learning 14](#_Toc112057807)

[Physics-informed machine learning: case studies for weather and climate modelling 14](#_Toc112057808)

[Application of sparse identification of nonlinear dynamics for physics-informed learning 15](#_Toc112057809)

[Recent machine learning applications in space 16](#_Toc112057810)

[Machine Learning for planetary rovers 16](#_Toc112057811)

[State of the art 16](#_Toc112057812)

[Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical system 16](#_Toc112057813)

[Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What’s next 16](#_Toc112057814)

[Machine learning in space 16](#_Toc112057815)

# Original papers given

## Discovering governing equations from data by sparse identification of nonlinear dynamical systems

* Original paper on the SINDy method
  + <https://www.pnas.org/doi/abs/10.1073/pnas.1517384113>
* Breaks down the maths

*So you set up the following sparse regression problem*

*You get your time series data, the matrix X, which you differentiate (there’s different ways to approximate the derivative), which is the matrix X dot. Then you build a library of candidate functions, which you sub in your time series data to get Θ(X). which is multiplied by the matrix Ξ, which has columns of sparse vectors, ξk, which tell us what terms are active in the library, which is was you solve for. This is an optimisation problem, and there are different optimisation methods you could use.*

* Goes through examples for systems where the governing equations are known
  + Chaotic Lorenz system
  + PDE for vortex shedding behind an obstacle
* Also discusses use for parameterised systems and externally forced systems
  + Include a bifurcation parameter in the dynamics
* Definitions
  + Attractor: variables in a system tend towards an attractor
  + Overfitting: model learns the noise
  + Regression: finding the relationship between independent and dependant variables

## SINDy-PI: A Robust Algorithm for Parallel Implicit Sparse Identification of Nonlinear Dynamics

* Update to the SINDy paper – improvement to allow for discovering PDEs, implicit equations
  + <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2020.0279>
* Discusses the theory
* Goes through the SINDy method
* Then the implicit-SINDy – but this is more of a stepping stone towards SINDy-PI
  + The problem now looks like
* Then details SINDy-PI
  + The idea is that if you know at least 1 term in the dynamics, then the problem can be written in a non-implicit form

*So we’re saying the term we know is equal to the library of terms, with the term we know removed, multiplied by a sparse vector which tells us which of the remaining terms in the library are active*

*To find this sparse vector, ξk, we solve*

*This is a non-convex optimisation problem, which can be relaxed. For example, using sequentially thresholded least-squares. The beta term is the sparsity promoting parameter (the greater the more sparse Xi is).*

*You find that first term by trying each term in the library until you get one that gives a sparse vector (this can be done in parallel hence SINDy-****Parallel*** *Implicit.*

* Model selection
  + The threshold lambda is a hyperparameter that’s tuned
  + To choose the best model you can use AIC and BIC [need to look into this further]
  + Or you could sweep through lambda values and choose the best
  + They choose lambda by using the test data to see what dives the lowest test error
  + Can also cross reference each model
* Constrained optimisation formulation
  + Instead of testing for each candidate function at a time you can test a single constrained system of equations

The Xi matrix has 0 entries on the diagonal b/c otherwise you would get a trivial solution. The sequentially thresholded least squares method 0s out any terms less that some parameter λ. We end up with multiple candidate models – 1 for each column of the Xi matrix. These models can be cross referenced / used to refine the library Theta.

* Noise robustness
  + SINDy-PI can handle significantly more noise than implicit-SINDy
* Data usage
  + SINDy-PI also uses much less data than implicit-SINDy
* Implicit PDE identification
  + SINDy-PI can be used to identify a PDE with rational terms – PDE-find cannot
* More examples
  + Mounted double pendulum
  + Single pendulum on a cart
  + BZ reaction
  + Extracting physical law from double pendulum
* Conclusions
  + SINDy-PI is a robust variant of the SINDy algorithm
  + Future work: designing the library, potential automatic library generation
* Definitions
  + Tensor: multi-dimensional array
  + A priori: independent from current experience
  + Loss function: tells us how close the current output is to the expected output

## Physics- informed machine learning

* Overview: general discussion on what physics informed machine learning is, mainly in the context of physics informed neural networks
  + <https://www.nature.com/articles/s42254-021-00314-5>
* Categories: small data lots of physics, some data some physics, big data no physics
* Physics can be embedded as observational biases
  + Like data from sensors
* Also, inductive biases
  + Embedding prior knowledge associated with a predictive task
  + Physics informed neural networks (PINNs) – embedding PDEs into the loss function
* Learning bias
  + Incorporating penalties
* Hybrid methods: can combine the above methods
* Benefits of physics-informed
  + Good for incomplete models and noisy data
  + Good for small amounts of data (strong generalisation)
  + Helps us understand deep learning
  + Good for high-dimensional problems
* Discussed the associated uncertainties
  + Due to the physics: in stochastic systems
  + Due to the data: noise
  + Due to the learning model: there’s limits to the models
* Gives some applications
  + Flow over an espresso cup (3D velocity and pressure fields)
  + 4D flow MRI (looking at fluid flow, applying the Navier-Stokes equations to denoise the data)
  + Edge plasma dynamics (learning turbulent field dynamics)
  + Etc
* There’s machine learning libraries available, a lot use python
* It depends on the specific task what model, framework, and algorithms to use
* The limitations
  + Doesn’t work for multiscale and Multiphysics problems
  + Training isn’t robust
  + It’s hard to get the right kind of data sometimes
  + We know it works empirically, but we don’t have the maths to explain it
* Future work
  + Digital twins
  + The same training data can produce different data-driven models – so we need to see how generalisable transformations are
  + Instead of humans deciding variables, these could be automatically determined from observations
* Definitions
  + Kernel: null space
  + In-vivo: in the body
  + Ab initio: from the beginning

## Simulation and Analysis of Vibration of Rolling Bearing

* **Physics model of vibration of a rolling bearing**
  + <https://www.scientific.net/KEM.588.257>
* Introduction
  + Rolling bearings are modelled linearly (Kelvin-Voigt) model
  + The non-linear model (due to the rolling elements exciting non-linear vibrations
* model of rolling bearing
  + outer race is fixed, inner race w/ mass attached rotates anticlockwise, so balls (massless springs) circulate clockwise.
  + The forces are:
    - The contact forces
      * Text, letter

        Description automatically generated
    - Damping force
      * A picture containing text

        Description automatically generated
    - Rolling friction force
      * 
  + Vibrations are described by differential equations of motion (in the y and the x)
    - Diagram, schematic

      Description automatically generated
    - A picture containing text

      Description automatically generated
* Simulation results
  + Looking at the clearance (l) increasing over time – the gap between the ball and the races increases due to wear
  + 3 amplitudes were looked at peak-peak, RMS, and average
  + From l = 0 to 100 um – periodic vibrations
  + From l = 100 to 204.8 um – significant & constant increase of amplitudes
  + For l > 204.8 um – jump of amplitude
  + Beyond this the vibrations are chaotic, look like noise
  + There’s a window of periodical vibrations from l = 240 to 320 um
* Analysis of vibrations
  + l < 204.8 um: periodical vibrations
  + 203.6 < l < 400 um: chaotic vibrations
    - Bistability interval
  + L > 203.6 um: windows of periodical vibrations
  + Decreasing for l < 239 um: period doubling bifurcation leading to chaos, the chaotic explosion
  + Increasing for l > 240 um: bifurcations leading to chaos (period doubling near l = 247.1 um), chaotic vibrations at l = 251.501 um – saddle node bifurcation
  + L = 328.5 um: periodical vibrations
  + L = 329.9 um: chaotic vibrations
    - Changing w/out noticeable reason
* Conclusion
  + Different phenomena were observed looking at the amplitudes of vibrations as functions of clearance
  + Very small changes in clearance cause big changes in the amplitudes
  + Non-linear models are better for rolling bearing vibrations

# Other SINDy variations

## Sparse identification of nonlinear dynamics for rapid model recovery

* [*https://aip.scitation.org/doi/full/10.1063/1.5027470*](https://aip.scitation.org/doi/full/10.1063/1.5027470)
* *Same group*
* *Abstract: for systems with abrupt changes – detects abrupt change, then applies SINDy to update an existing model (rather than create one)*
* *Comes with python code*
* *2018*

063116-3 
FIG. of abrup,.srNDy 
in IV A 
ii(x) = 
Deletion 
Addition 
Chaos 28, 063116 (2018) 
- -2.672 + I.oory 
Change 
— —10.00r+ 10.00" 
- -2.67: + I.ooxu 
SINDy models obtained via 
sparse regress ton 
Combinati on 
"f singlc in i" 
Midition. deletim_ a data fM ax 
Where Oi(x) form a Set of nonlinear candidate functions. The 
candidate functions may be chosen 10 be polynomials, trigo- 
nometric functions. or a more general Set Of functions. 
With poor choice of the candidate functions i.e., if library 
functions are non-orthogonal and/or overdetermined. the 
SINDy approach may fail to identify the correct model. 
Sparse models may be identified from time-series data, 
which are collected and formed into the data matrix 
(3) 
We estimate the time derivatives using a simple forward 
Euler finite-difference scheme, i.e., the difference of two 
consecutive data, divided by the time difference 
This estimation procedure is numerically ill-conditioned if 
data are noisy. although there are many methods to handle 
noise which work very well if used correctly. 
Noise- 
robust derivatives were investigated in the original SINDy 
algorithm Next, we consider a library of candidate nonlin• 
car functions O(X) of the form 
sin (X) 
Here, the matrix X" denotes a matrix with column vectors 
given hy all possible time-series of d-th degree polynomials 
in the State x. The terms in can be functional forms moti- 
vated by knowledge Of the physics. Within the proposed 
work, they may parameterize a piecewise-affine dynamical 
model. Following best practices of statistical learning; to 
preprcxess. wc mean-subtract and normalize each column of 
to have unit variance. The dynamical system Can be 
represented in terms of the data matrices as 
"Hue coefficients in the column of determine the active 
terms in the k•th row of Eq- (2). A parsimonious model has 
the fewest terms in required to explain the data. One 
option to obtain a sparse model is via convex El-regularized 
regression 
The hyper parameter balances complexity and sparsity 
of the solution. Sparse regression. such as least absolute 
shrinkage and selection operator (LASSO)" and sequential 
thresholded least-squares, improves the robustness of iden- 
tiheation for noisy overdetermined data. in contrast to earlier 
methods using compressed sensing. 
61.62 Other regulariza- 
lion schemes may be used to improve performance, such as 
the elastic net regression. 

* SINDy 🡪 detect change 🡪 SINDy again to update model
* For the sparse regression step in SINDy – sequentially thresholded ridge regression (rather than sequentially thresholded least squares
  + *Will look in detail at what this is later if needed*
* Types of abrupt changes
  + Variation of a term: least squares regression to see what terms have changed
  + Deletion of a term: SINDy regression on the active terms – to see what terms have dropped out
  + Addition of a term: SINDy regression on the inactive terms (of the baseline model)
* Step A: baseline SINDy model
  + Uses gridsearch to find the best hyperparameter
    - All combinations of hyperparameters are tested and the best is picked
      * α - ridge regression regularisation parameter
      * γ – the thresholding parameter
      * ndegree­ – the max. degree of the polynomial feature transformation [?]
      * n­fold – number of cross-validation runs [?]
* Step B: abrupt change is detected by looking at the Lyapunov time of the data with the model prediction
  + Lyapunov time: a timescale after which the dynamic system is chaotic
  + [*http://fy.chalmers.se/~f99krgu/dynsys/DynSysLecture10.pdf*](http://fy.chalmers.se/~f99krgu/dynsys/DynSysLecture10.pdf)
* Step C: apply SINDy regression to update previous model
  + Look at the new data in the existing sparse model
  + Sparse regression on terms active in the baseline model
  + If there’s still an error, sparse regression on inactive terms
  + Repeat until error is small enough / no error (convergence)
* Used Lorenz system and Van der Pol oscillator as examples
* Noise and data volume
  + Abrupt-SINDy is more accurate (than SINDy) for short time
  + Works for larger data sets
  + Better (than SINDy) for noise
* Limitations
  + Instead of using the baseline model, dynamic mode decomposition as alternative model
  + Hyperparameterisation: could change fixed hyperparameters to be learned instead
  + Lyapunov time: change how this is estimated
  + Could use other optimisation techniques

## SINDy with Control: A Tutorial

* [*https://ieeexplore.ieee.org/abstract/document/9683120*](https://ieeexplore.ieee.org/abstract/document/9683120)
* *Same group*
* *Abstract: using SINDy with Model predictive control optimisation, uses infectious disease example*
* *Comes with matlab code*
* *2021*
* SINDy w/ control –
  + **Text

    Description automatically generated**
  + Where x is the state and u is input terms

**Diagram

Description automatically generated**

* Model Predictive Control (MPC)
  + For strongly non-linear systems w/ constraints, time delays, non-minimum phase dynamics, and instability
  + *Will go over in more detail if needed*
* Tutorial on infectious disease
  + Will go through example (MATLAB code)
* Can change assumptions: the differentiation method, the library functions, the sparse regression algorithm, the value of the sparsity-promoting hyperparameter
* And change the forcing functions, and control strategies
* Definitions
  + AIC and BIC – methods for model selection

### AIC and BIC

Methods for model selection <https://machinelearningmastery.com/probabilistic-model-selection-measures/>)

Statistical or probabilistic model selection: combining the complexity of a model with the performance of the model into a score and selecting the model that optimises the score

* Performance = how well a candidate model has performed on the training dataset
  + Evaluated using a probabilistic framework (log-likelihood)
* Complexity = how complicated the trained candidate model is
  + Look at no of degrees of freedom or no. of parameters

Pros: dont need test dataset

Cons: the statistic is unique for each model, dont account for uncertainty

Log-likelihood comes from maximum likelihood estimation

AIC : Akaike Information Criterion

* + N is no. of training examples, LL is log-likelihood, k is no. of parameters
* Want to minimise this (meaning smallest AIC is the best model)
* AIC may choose more complex models that have good performance (whereas BIC penalises complexity more

BIC: Bayesian Information Criterion

* + Variables same as for AIC
* Again, this is minimised
* It is proportional to AIC
* BIC will give more complex models a greater score
* If one of the candidate models is the true model, then the greater the size of the training data, the more likely BIC will choose it

AICc: second order AIC

* Basically just AIC w/ an extra penalty term
* (the k sqaured term is what makes it 2nd order)
* Used for a small sample size
  + b/c AIC can overfit (ie select a model w/ too many parameters)
* Pros - Tends to be more accurate
* Cons - Can be more difficult to compute

## SINDy-SA: Enhancing nonlinear system identification with sensitivity analysis

* [*https://assets.researchsquare.com/files/rs-1317964/v1\_covered.pdf?c=1644418475*](https://assets.researchsquare.com/files/rs-1317964/v1_covered.pdf?c=1644418475)
* *Different group*
* *Abstract: so we don’t need to manually choose the threshold*
  + *Discusses predator-prey model*
* *Comes with python code (and uses gnuplot)*
* *2022*
* Previously, to choose the threshold, you would pick the best model using each threshold in some predefined set
* Sensitivity analysis classifies the parameters according to importance
* SA method: Morris a.k.a. Elementary Effects (EE) method
* SINDy-SA steps
  + Ridge regression
    - **Text

      Description automatically generated**
    - Where changing alpha controls the impact of each term, giving different models
  + Error computation
    - Working out the error [sum of squared errors (SSE)] between the measured derivatives and the approximated
    - SSE at each iteration is compared to the last, repeating until the SSE increases significantly
  + Sensitivity analysis
    - Works out ‘sensitivity indices’ so that the xi terms can be ranked most to least influential
* Framework

A picture containing graphical user interface

Description automatically generated

* Model selection methods
  + Akaike Information Criterion (AIC) – estimates prediction error and quality of models relative to others
  + Second order AIC (AICC) – used for small sample sizes
  + Bayesian Information Criterion (BIC) – for choosing from a finite set of models, based on the likelihood function, want low BIC
* Specific methods used for the steps in the examples of SINDy-SA given
  + To generate the simulation data: LSODA
  + To approximate the derivatives: second-order finite difference method
  + The optimisation: Levenberg-Marquardt optimisation algorithm
  + Model selection: pareto curve & calculation AIC, AICC, BIC.
* Predator-prey model: emphasises the recalibration step [?]
* Tumour growth model: shows there’s a limit on thresholding
* Pendulum motion model
* (epidemiological SIR) Compartmental model

## Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control

* [*https://royalsocietypublishing.org/doi/full/10.1098/rspa.2021.0904*](https://royalsocietypublishing.org/doi/full/10.1098/rspa.2021.0904)
* *Same group as other SINDy papers*
* *Abstract: gets a few different options from using SINDy and then gives a probability of what dynamics are in the system, can handle a lot of noise, uses Lotka-Volterra equations as an example*
* *Comes with python code (in the pysindy github)*
* *2022*
* Classical statistical bagging methods
* Introduction
  + Gives background – listing different data-driven model discovery methods
  + Briefly explains SINDy and what its been used for
  + Different methods for dealing w/ noise are ensemble methods (other works have investigated ensemble methods)
  + Need to look at the uncertainties
* Definitions
  + Bagging: bootstrap aggregation
    - Bootstrap method: A technique for estimating by average estimates from multiple small data samples
    - You select a random sample and replace it (meaning it could be selected again)
    - Can do this multiple times with the data
    - <https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/> , <https://machinelearningmastery.com/essence-of-bootstrap-aggregation-ensembles/>

# Other papers related to rolling bearings

## A Deep Learning-based Approach for Fault Diagnosis of Roller Element Bearings

* pybearing
* <http://www.me.sc.edu/research/downey/publications/Conference_publications/Sadoughi_2018_Deep_Learning_based.pdf>
* A convolutional neural network based approach
* The approach has 2 steps
  + Pre-processing (spectral kurtosis) the raw sensor data
  + Then feeding these signals to a CNN (1D multichannel)
* [sec. 1 intro] fault detection = save money , roler bearings have high failure rates
* Raw sensor data needs pre processed
* Deep learning can automatically detect and learn
* A shallow CNN has been used for machine fault analysis before
* Using a lot of data with low noise is achievable and works
* Unique to SCNN 🡨 the name of the approach detailed in this paper
  + Multiple sources of information
  + The bearing fault is recognised separately from other malfunctions
  + Filtering improves SNR
* [sec. 2 CNN] CNNs are multistage neural networks
  + *will maybe look at in more detail from another source to understand better*
* [sec 3. SCNN] raw signal 🡪 spectral kurtosis analysis 🡪 envelope analysis 🡪 frequency domain transformation 🡪 signals combined 🡪 CNN
* [sec 4. Experiment] 2 bearings on a simulator, 4 types of faults looked at
  + Inner race defect
  + Outer race defect
  + Ball defect
  + Combination of above
* Gathered 2340 measurement samples
* The SCNN was implemented
  + *Again come back to for more detail*
* Of 8 samples, 7 used to train and 1 used to test – SCNN was accurate (more so than other machine learning and CNN approaches)

## Nonlinear model and simulation of a rolling bearing

* <https://iopscience.iop.org/article/10.1088/1757-899X/710/1/012006/meta>
* Note: not read in depth, just picked out equations

## Effect of the raceway defects on the nonlinear dynamic behavior of rolling

* <https://link.springer.com/content/pdf/10.1007/s12206-019-0501-0.pdf>
* Note: not read in depth, just picked out equations

# Other PIML methods

## Discovering unmodeled components in astrodynamics with symbolic regression

* Symbolic regression
* <https://strathprints.strath.ac.uk/72508/1/Manzi_Vasile_IEEE_WCCI_2020_Discovering_unmodeled_components_in_astrodynamics.pdf>
* Definitions
  + Epochs: Astronomy - an arbitrarily fixed date relative to which planetary or stellar measurements are expressed. (time periods)

Python toolkit

* Github: <https://github.com/trevorstephens/gplearn>
* Read the docs: <https://gplearn.readthedocs.io/en/stable/intro.html#intro>

# Space applications

## Incorporating Physical Knowledge Into Machine Learning for Planetary Space Physics

* [*https://www.frontiersin.org/articles/10.3389/fspas.2020.00036/full*](https://www.frontiersin.org/articles/10.3389/fspas.2020.00036/full)
* *Looks at PIML planetary space physics*
* Uses data from Cassini – orbited Saturn, first large scale data collection
  + Spatial-temporal data (from plasma and wave sensors [?])
    - Intensity of ions each with a spacecraft position and time dependance
  + Used for predicting weather
* In the middle of the arrow diagram – physics based, data driven
* Framework (interpretable semi-supervised event detection)
  + Limit region of interest
  + Careful consideration of training and test datasets
  + Normalise and/or transform
  + Incorporate physical calculations - like summing/ integrating
  + Use different penalties
  + Standardising event definitions
  + Try different ML models and datasets
    - This used logistic regression and random forest classification (both supervised)

## Science Autonomy and Space Science: Application to the ExoMars Mission

* [*https://www.frontiersin.org/articles/10.3389/fspas.2022.848669/full*](https://www.frontiersin.org/articles/10.3389/fspas.2022.848669/full)
* *Talks about ML in a mars mission*
* Talks about the idea of science autonomy
  + Where an instrument can operate on its own and work out the most important data from what it collects on its own
  + Minimise ground-to-space interactions and maximise science return
* Used for the ExoMars mission to search for signs on life – using the Mars Organic Molecule Analyzer
* Input: a database of lots of known mass spectra
* Analysis process
  + Preprocessing: picks best samples, converts to a dataset that can be used
  + Filtering: unsupervised ML – separates data into clusters
  + Matching: supervised ML – uses a “customer’s also liked” algorithm to compare samples to database
    - This stage is done by a person choosing a sample and a database – but full autonomy would been this happening onboard
* Limitations/challenges
  + data volumes
    - can use data augmentation e.g. change known good quality data a little bit to generate new data that can be used to train the ML models
    - can use transfer learning
  + technology – risk of radiation/other extreme conditions ruining the data

## Machine learning in space: extending our reach

* <https://www.webofscience.com/wos/woscc/full-record/WOS:000293297200003>
* *Abstract talks generally about applying machine learning to space missions (note its out of data (published 2011))*
* The nature of space missions leads itself to favouring autonomy, both for missions carried out remotely and on the ISS
* Challenges: the computational ability is lost when the equipment needs to be protected against radiation; the technique has to be proven to be reliable
* Existing examples of ML and AI in space: Mars Rovers autonomous navigation and image analysis (chooses what images to be sent to ground)
* The effect of high radiation on the ML algorithms – the k-means algorithm can withstand radiation whereas the kd-k-means is more sensitive
* Lots of options for uses of ML in space: image analysis, time series analysis (fault detection/prediction), classification, clustering etc
* References another paper (<https://link.springer.com/content/pdf/10.1007/s10994-011-5239-6.pdf>) that discussed crater identification

## Automated Crater Detection on Mars using Deep learning

* <https://www.sciencedirect.com/science/article/pii/S0032063318303945>
* NN for detecting craters on Mars
* Used an automated crater detection algorithm (CDA) based on a CNN to ID circular crater like feature from a digital terrain model (DTM)
* Uses CNN trained on lunar surface data – transfer learning, a CNN trained on Mars data and another CNN trained on Mars data that looks for disks (rather than circles, aka the rim of the crater)

## Identifying Exoplanets with Deep Learning

* <https://iopscience.iop.org/article/10.3847/1538-3881/ab0e12/meta>
* Modified CNN using data from NASA’s Kepler mission
* Takes light signals from the K2 data (data from the Kepler telescope after a flywheel failure)
* Had to label >30k events by hand (b/c using supervised learning)
* Doesn’t work – IDs too many false positives

## Physics-informed machine learning: case studies for weather and climate modelling

* <https://royalsocietypublishing.org/doi/epdf/10.1098/rsta.2020.0093>
* 10 case studies on different approaches to PIML
* Intro
  + We think about PIML because some of the disadvantages to ML is it doesn’t always obey the underlying physical principles of the system it is applied to
  + Sometimes they can learn the physics, but then it wouldn’t transfer
  + And you need a lot of good quality data
* Objectives of PIML
  + Better predictive models
  + More data efficient
  + Speed up training
  + Make models more generalisable
  + More interpretable
* 10 approaches
  + 1 physics based regularisations
    - Aka custom loss functions – add physics terms to the loss function
  + 2 custom NNs
    - add physical constraints
  + 3 symmetries, invariances, equivariances (geometric properties of spatio-temporal dynamics)
    - E.g. rotational symmetry == conservation of ang mom
  + 4 stochasticity
    - To accurately represent chaotic and turbulent systems
  + 5 stability
    - Consider the physics when choosing the inputs and outputs can achieve stability
  + 6 multi-scale properties
    - Consider that changes at a small scale & high freq impact the system on a large scale & low freq
    - Can be done through spectrum and covariance function
  + 7 spatio-temporal coherence
    - Use the systems governing equations
  + 8 physics-based modelling frameworks
    - Take physics models and use ML to replace approximations and empirical parameters
  + 9 interpretability
    - Is domain specific
  + 10 uncertainty quantification
    - Accounts for noisy and low quality training data
* Applications of PIML in weather and climate modelling
  + Emulating complex processes that aren’t understood/ well represented
  + Downscaling coarse data 🡪 hi-fi, high-res data
  + Forecasting spatio-temporal dynamics

## Application of sparse identification of nonlinear dynamics for physics-informed learning

* <https://ieeexplore.ieee.org/abstract/document/9172386>
* Has a couple examples of someone testing SINDy using synthetic data
* Modified van der pol oscillator
  + Mentions how to find best sparsification parameter
    - Ran for 28 values between 0 and 1 and used AIC to find the best one
* Stochastic system
  + Needed multiple trajectories (ran 1500)

## Recent machine learning applications in space

* <https://ieeexplore.ieee.org/abstract/document/9132719>
* Discusses exoplanet detection and anomaly detection is spacecraft telemetry
* Exoplanet discovery (from Kepler light curves)
  + Citizen science – people look at the data to manually detect anomalies
  + Comes with a jupyter notebook: <https://github.com/spacetelescope/notebooks/blob/master/notebooks/MAST/Kepler/Kepler_Lightcurve/kepler_lightcurve.ipynb>
* Spacecraft telemetry
  + Uses a GAN (generational adversarial network)

## Machine Learning for planetary rovers

* <https://www.sciencedirect.com/science/article/pii/B9780128187210000197>

# State of the art

(Didn’t end up using in presentation)

## Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical system

* Low-dimensional models for high-dimensional systems
* <https://arxiv.org/pdf/1912.08177v5.pdf>

(For the NN example)

## Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What’s next

* Literature review on PINNs (published 06/2022)
* <https://arxiv.org/abs/2201.05624>

NOTES ON ARTICLES

## Machine learning in space

<https://eos.org/opinions/ten-ways-to-apply-machine-learning-in-earth-and-space-sciences>