# Tutorial: Machine Learning and Artificial Intelligence in Batteries

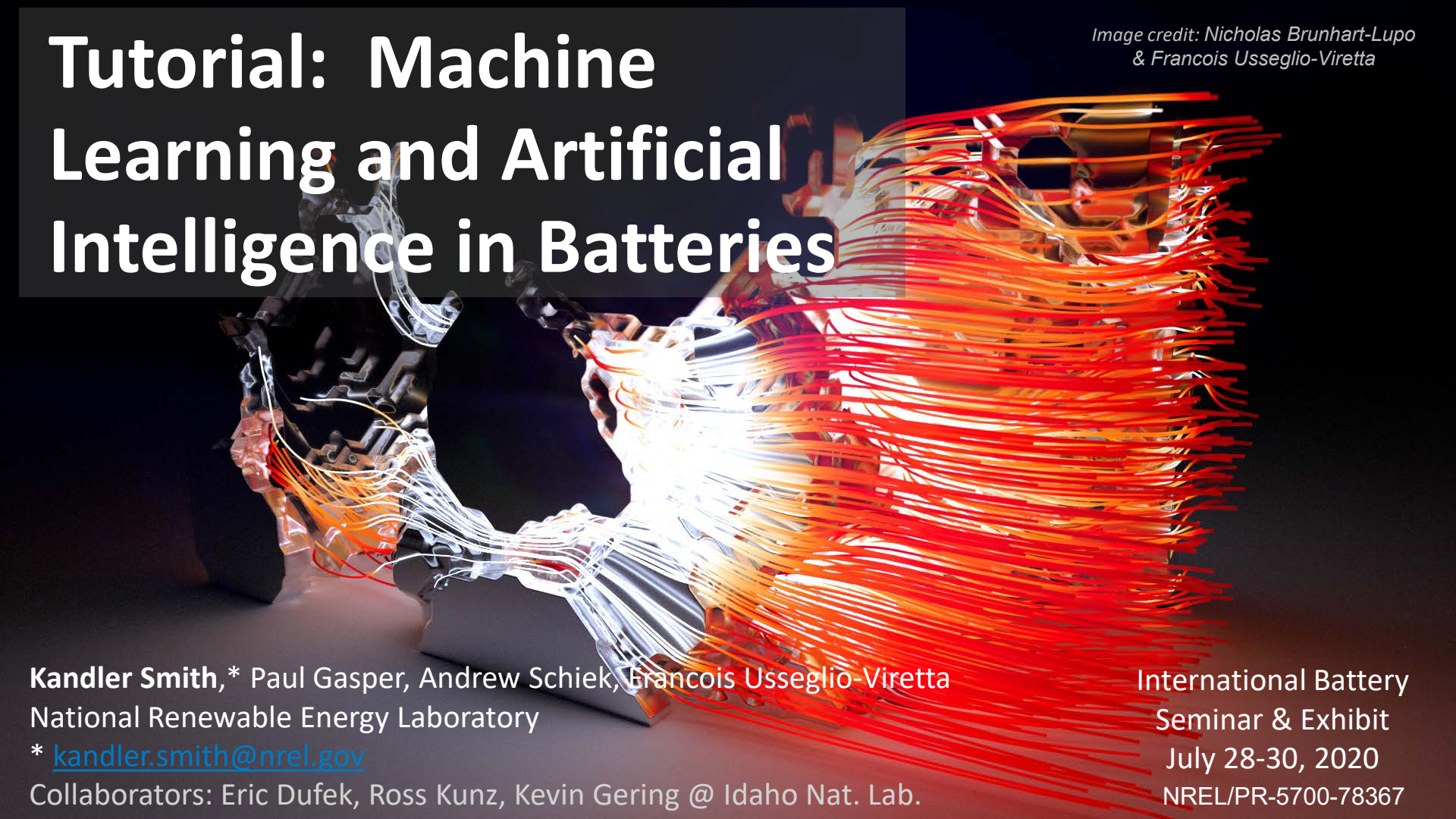


*Image credit: Nicholas Brunhart-Lupo & Francois Usseglio-Viretta*

**Kandler Smith**,* Paul Gasper, Andrew Schiek, Francois Usseglio-Viretta
National Renewable Energy Laboratory
* kandler.smith@nrel.gov
Collaborators: Eric Dufek, Ross Kunz, Kevin Gering @ Idaho Nat. Lab.

International Battery Seminar & Exhibit
July 28-30, 2020
NREL/PR-5700-78367

# What is Machine Learning?

- "Gives computers the ability to learn without being explicitly programmed" (Arthur Samuel)
  - Artificial intelligence
- Matching physical models to data (My interest)
  - Automating my job
  - Accelerate experiments, designs, applications, discovery
    - Gaps → learn new battery physics
- **Concepts**: Optimization/regression with cost functions involving hyperparameters; Cross validation; Hierarchical models

# Contents

# PART 1

Introduction & Simple Algorithms

Manufacturing Process Example

# Part 1 – Introduction & Simple Algorithms

- Introduction
  - Resources
  - Types of Algorithms
    - Unsupervised vs. Supervised
  - Motivating Battery Examples
- Simple algorithms
  - K-means clustering
  - Decision tree

# Resources

Echem & Materials Disciplines

- Electrochemical Society Data Sciences Hacks
- Materials Research Society tutorials
- U. Maryland Bootcamp
  - http://nanocenter.umd.edu/events/mlmr-2020/
  - Materials Research, Microscopy data

## General Theory

- https://towardsdatascience.com, Wikipedia
- Coursera.org – Machine Learning (Andrew Ng)
  1. **Linear regression** with multiple variables
     1. Matlab, Octave
  2. **Logistic regression** – classification, hypothesis, decision boundary, **cost function, gradient descent**
     1. Regularization, multiclass classification (one-vs-all)
  3. **Neural Networks** – speech, image recognition, …, non-linear regression
     1. How the brain works
  4. Neural Networks Learning – backpropagation algorithm
  5. Best practices – **Train/Validation/Test datasets**, Should I collect more data?
     1. Bias (underfitting) vs. Variance (overfitting); Learning Curves; Skewed data
  6. Support Vector Machines
  7. Unsupervised Learning – market segmentation, text summaries
     1. **Principal Components Analysis, K-Means**
  8. Anomaly Detection – e.g. fraud detection, manuf. Outliers
     1. Recommender systems – collaborative filtering, low-rank matrix factorization
  9. Large-Scale ML – Stochastic & Mini-Batch Gradient Descent; Parallelization
  10. **Image Recognition** Examples – recognize objects, words in an image, facial recognition
      1. **Artificial data synthesis** (+meaningful noise) vs **Crowd source**
      2. Algorithm pipeline

# Additional Background Resources

- ML in Materials Science – *Excellent description of many algorithms*
  - **T. Mueller, A.G. Kusne, R. Ramprasad, "Machine Learning in Materials Science," Reviews in Computational Chemistry, 29: 186-273 (2016)**
- ML Theory
  - G. James, D. Witten, T. Hastie, R. Tibshirani, "An introduction to statistical learning" 2013.
  - J. Friedman, T. Hastie, R. Tibshirani, "The elements of statistical learning" 2010.

# Computational Resources

Languages

- Matlab (engineers - $), Octave (free version)
- R (statisticians)
- Python 3.7 (everybody)
  - https://www.anaconda.com/distribution/#download-section

Libraries

- NumPy, SciPy, Scikit-learn (regression, clustering, …)
- PyTorch (computer vision, natural language processing – Facebook AI)
- TensorFlow (deep learning neural networks - Google Brain Team)

# Types of Algorithms

- Supervised vs Unsupervised
- Continuous and/or Discrete (Classification)

# Supervised learning

- Algorithm is told what is the correct answer
  - Labeled input/output training data (x,y)
- Regression problem (**Continuous**): Fitting a function to data
  - Feature selection
  - Feature scaling important
- Classification problem (**Discrete**): Classification problem – is mechanism present or not, using infinite number of features/attributes
  - Logistic regression
  - Support Vector Machine

Figure: T. Mueller, Reviews in Comp. Chem. (2016)

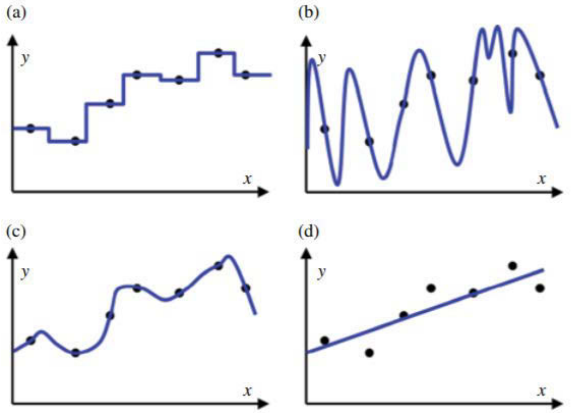
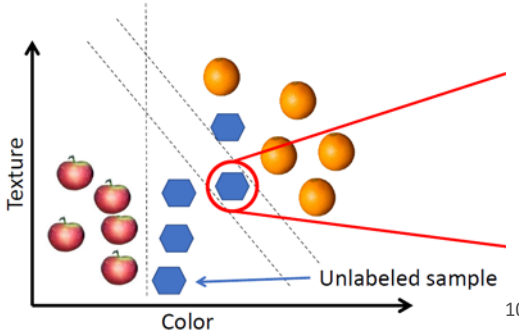



Figure:
A. Gilad Kusne
Daniel Samarov



# Unsupervised learning

- Algorithm is not told the correct answer
- Tries to determine “categories” or “types” from data
- Needs only input data (x)

Examples:

- Google news sorting
- Deconvolution
- Market segmentation

Figure: Andrew Ng

# Supervised vs Unsupervised Learning

## *Applications in Materials Science*

**TABLE 1** Supervised and Unsupervised Learning Examples

| | Example Methods | Selected Materials Applications |
|---|---|---|
| Supervised learning | Regularized least squares<br>Support vector machines<br>Kernel ridge regression<br>Neural networks<br>Decision trees<br>Genetic programming | Predict processing structure–property relationships; develop model Hamiltonians; predict crystal structures; classify crystal structures; identify descriptors |
| Unsupervised learning | *k*-Means clustering<br>Mean shift theory<br>Markov random fields<br>Hierarchical cluster analysis<br>Principal component analysis<br>Cross-correlation | Analyze composition spreads from combinatorial experiments; analyze micrographs; identify descriptors; noise reduction in data sets |

Table: T. Mueller, Reviews in Comp. Chem. (2016)

# Motivating Battery Examples

# Applications of ML to Batteries

1. Manufacturing quality
2. Microscopy/image processing
3. Echem data (IVT) → Lifetime, performance
4. Materials research
5. Other battery R&D (safety)

**Algorithms**

- Clustering
- Neural Networks
- Penalized Regression
- Gaussian Process Regression

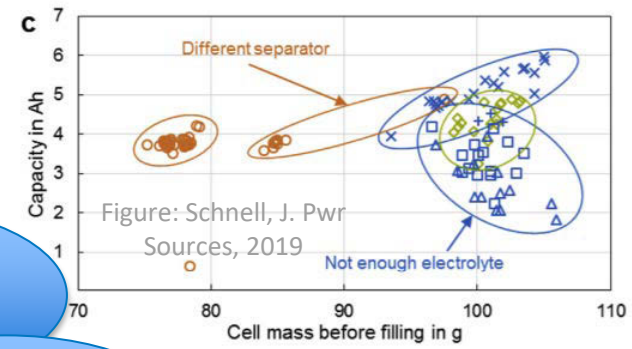


Figure: Schnell, J. Pwr Sources, 2019

**Data**

"Crowd-sourced" data (Andrew Ng)
… for Batteries:

- Real-world/complex data
- Density functional theory simulations
  - Materials Project Database
- 3D continuum simulations
  - Echem/thermal
  - Microstructure

# Simple Algorithms

(U) K-means Clustering (discrete) + Example

(U) Principal Components Analysis (continuous/linear algebra)

(S) Generalized Linear Model (continuous)

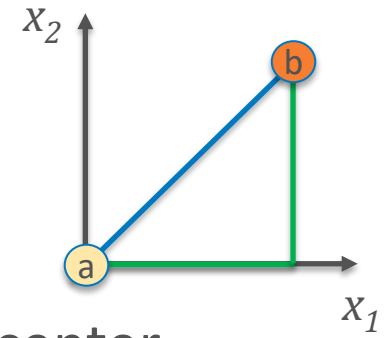
(S) Decision Tree (discrete) + Example

# K-Means Clustering: Overview and Goals

- Accurately group data using the fewest number of cluster centers (k) as possible
- Provides insight of potential patterns within data and makes analysis easier
- Simple unsupervised method (input data doesn’t need to be labeled)

# K-Means: How it works

- Based on the number of cluster centers, the nearest points will be sorted to the nearest cluster center.
- Two common methods used to define distance:*
  - Euclidean or $L_2$ norm $\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}.$
  - Manhattan City Block, Taxi Cab or $L_1$ norm $\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$




- Algorithms will typically iterate through different center locations, $c_j$, to find the optimum cluster centers while the user sets the number of cluster centers to be used

*https://en.wikipedia.org/wiki/Norm_(mathematics)

# K-Means: Algorithm Description

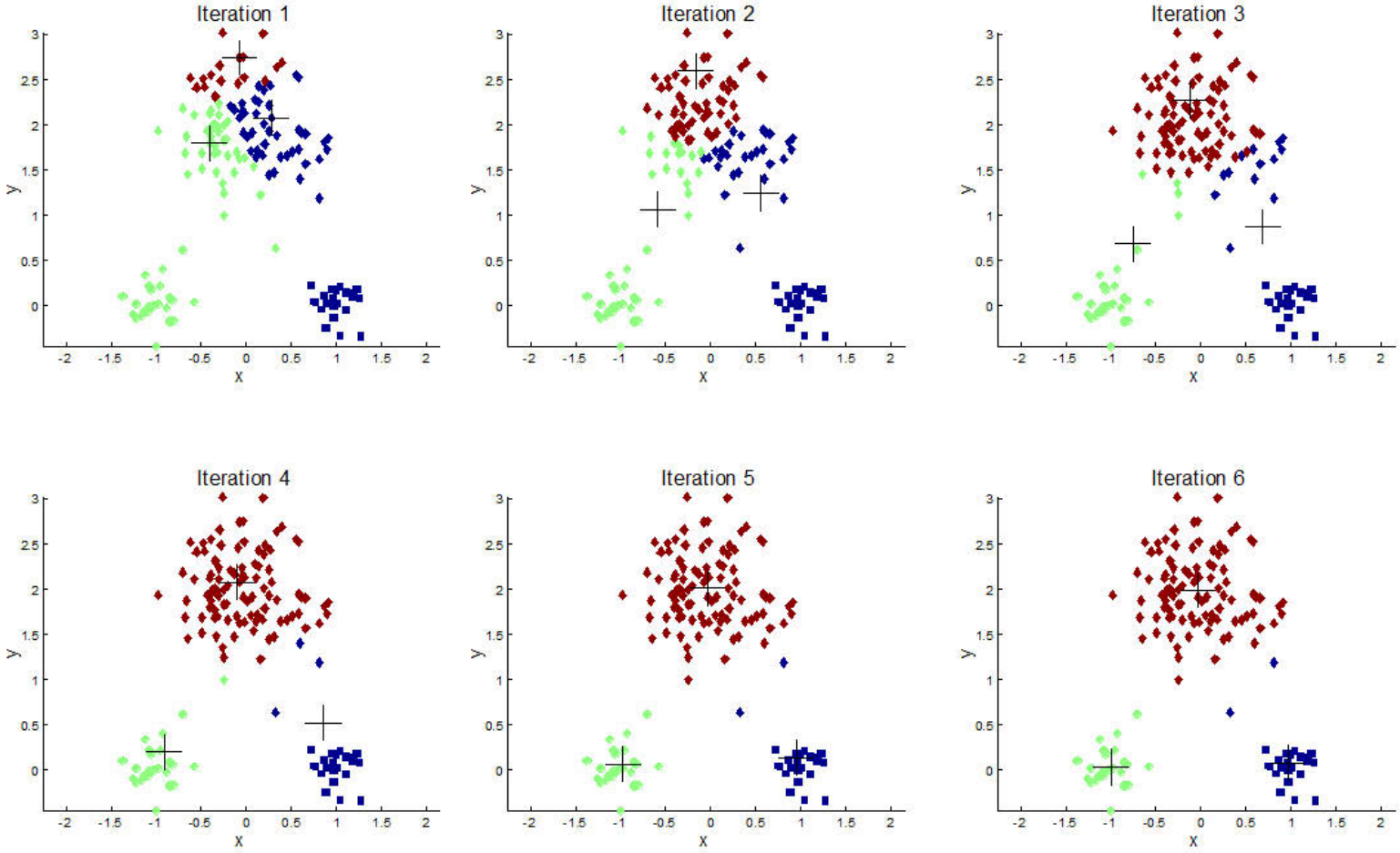
- Both the number of centers and the initial center location needs to be chosen by the user.
  - The initial center locations should be fairly spread out
  - Pre-clustering algorithm[3] can help determine number
- The distance to the respective centers will then be calculated and the point selected will belong to the cluster whose center it is closest too.
  - If we consider point p ($x_1$, $y_1$) and centers $c_1$ ($x_2$, $y_2$) and $c_2$ ($x_3$, $y_3$). Then using the Euclidean method for distance p will be assigned to $c_1$ or $c_2$ based on:

$$\min\left(\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}\right)$$

# K-Means: Algorithm Description, cont.

- After all values have been sorted to a cluster, the center of each cluster is calculated.
  - $C_{i,x} = \frac{(\sum_{i=1}^{N} x_i)}{N}, C_{i,y} = \frac{(\sum_{i=1}^{N} y_i)}{N}$
- After the new cluster centers are calculated, points are reassigned to the new cluster centers.
- This process repeats itself until no values change clusters.

# K-Means: Generic Example



# K-Means: Evaluating Clusters

- Need to balance accuracy vs. simplicity
- Accuracy of a cluster is simply defined as:
  - $Accuracy = \frac{Most\ Frequently\ Occuring\ Class\ in\ Cluster}{Total\ Instances\ in\ Cluster}$

    (Requires hand-annotated “truth” data)

- User must decide what change in accuracy is significant enough to them to add an extra cluster center

# K-Means: Importance/Uses

- Accuracy similar to that of more robust algorithms while being less computationally expensive.[2]
- Due to unsupervised nature can be useful when new data is coming in
- Allows us to gain information about data and assumptions prior to further analysis or model making.

# References

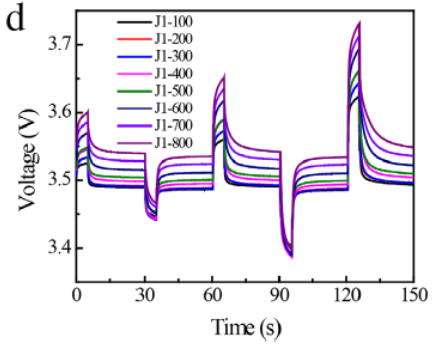

1. Forster, E.M., et al. “Cluster Analysis: See It 1st.” *Data Visualization*, 7 Aug. 2011, apandre.wordpress.com/visible-data/cluster-analysis/.
2. “K-Means Clustering.” *IBM Knowledge Center*, www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/r_kmeans_clustering.html.
3. Andrew McCallum, Kamal Nigam, and Lyle Ungar, “Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching,” 2000.


Programming examples for k-means clustering in Python.

- https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1?gi=572a98f4e093
- https://blogs.oracle.com/datascience/introduction-to-k-means-clustering

# K-means Clustering Battery Example

(A. Ran, Adv. Theory & Sim., 2020)





FULL PAPER

ADVANCED THEORY AND SIMULATIONS

www.advtheorysimul.com

## Data-Driven Fast Clustering of Second-Life Lithium-Ion Battery: Mechanism and Algorithm

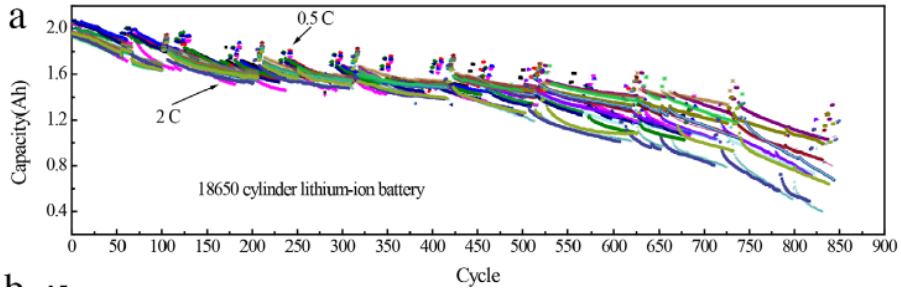
*Aihua Ran, Zihao Zhou, Shuxiao Chen, Pengbo Nie, Kun Qian, Zhenlong Li, Baohua Li, Hongbin Sun, Feiyu Kang, Xuan Zhang,* and Guodan Wei**

While electrical vehicles (EVs) are expanding rapidly and getting more and more popular in the market, researchers have started to leverage the remaining capacity of used or to-be-retired batteries for their second-life applications. It is crucial to develop a fast and efficient technology to first sort them and then extend their life while delivering energy, waste reduction, and economic benefits. In this work, a pulse clustering model embedded with improved bisecting K-means algorithm is developed to effectively sort retired batteries with life cycles ranging from new to an end-of-life state. The relevance of selected variables is rigorously validated, reaching the accuracy as high as 88% compared with the traditional full charge–discharge test. To note, the test time has largely reduced from hours to minutes. This data-driven clustering modeling with fast pulse test is a promising approach for clustering lithium-ion batteries, which is demonstrated with a home-built and high throughput intelligent clustering machine. In general, the technology opens a new generation of battery clustering, improving the efficiency and accuracy over the past semiempirical approaches.

### 1. Introduction

Lithium-ion batteries (LIBs) have a very wide range of applications due to their low price, decreasing cost, high energy density, and long lifetime.[1,2] However, owing to the current manufacturing technology, batteries currently used in electric vehicles (EVs) do not maximize the life of LIBs.[3,4] Generally, a large proportion of batteries retired from EVs have about 80% capacity of their initial capacity.[5,6] Accompanying with aging, LIBs experience voltage decay, resistance increase and capacity loss, causing the obvious difference among retired batteries.[7] Taking battery thermal management as an example, different degrees of battery aging will increase the difficulty of thermal management and increase the risk of battery use.[8–10] Therefore, accurately clustering the usable batteries with

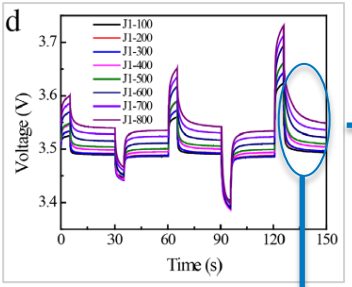
Rapid screening test to estimate C/5 full capacity (10-hour test!) in 3.5 minutes.



# K-means Battery Example (cont.)

(A. Ran, Adv. Theory & Sim., 2020)

Rapid pulse test @ 5% SOC




Principal Components Analysis (PCA)

Finds dominant eigenvalues/modes

Feature 1

Fitting of relaxation data

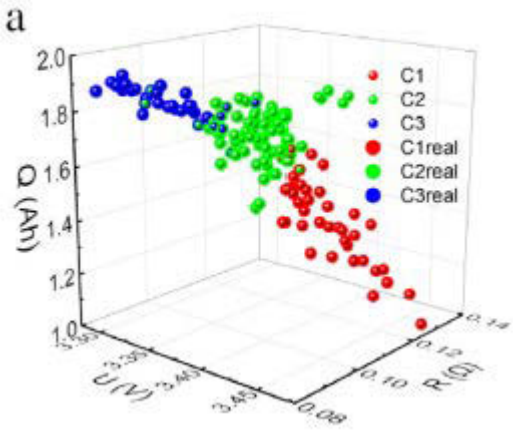
$V = \mathbf{U} + \mathbf{R}*\exp(-t/\tau)$

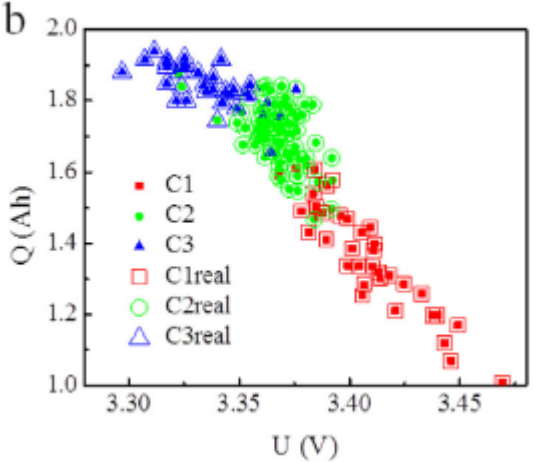
**U**

**R**

K-means Clustering

Good, moderate, poor cells



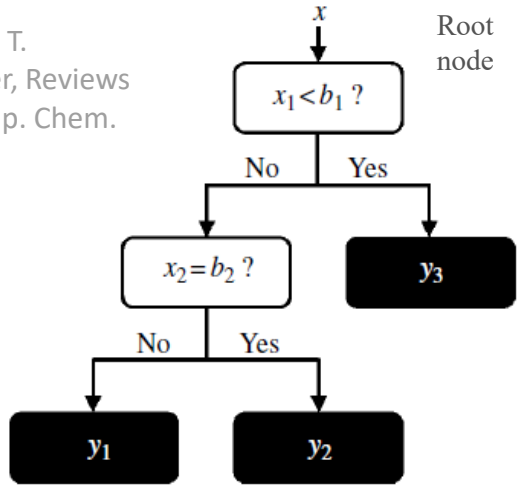
# Generalized Linear Model

- Supervised, continuous algorithm
- Linear regression generalized to account for response variables that have non-normal error distribution
  - Probability distributions: normal, binomial, Poisson, gamma, …
  - Unifies linear regression (continuous values), logisitic regression (discrete values), Poisson regression
  - Algorithm: Iteratively reweighted least squares for maximum likelihood cost function

# Decision Trees

- Supervised, discrete learning algorithm
- Top-down recursive method
- Attributes are either categorical or discretized numerical data
- Data are split at each level based on the attribute's value
- Hunt's Algorithm: Grows decision tree recursively by splitting training data into purer subsets
- Stop splitting at either set number of nodes or when a "leaf" becomes completely pure.

Figure: T. Mueller, Reviews in Comp. Chem. (2016)




An example of a simple decision tree. The leaf nodes are in black.

*Example use: Predict material properties from DFT database (melting point, density, conductivity,…)*

# GLM & Decision Tree – Manuf. Example

(J. Schnell, J. Power Sources, 2019)

ELSEVIER

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Data mining in lithium-ion battery cell production

Joscha Schnell*, Corbinian Nentwich, Florian Endres, Anna Kollenda, Fabian Dis
Thomas Knoche, Gunther Reinhart

*Technical University of Munich, Institute for Machine Tools and Industrial Management, Boltzmannstr. 15, D-85748, Garching, Germany*

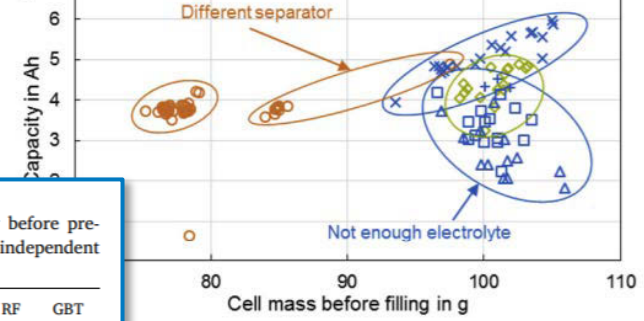
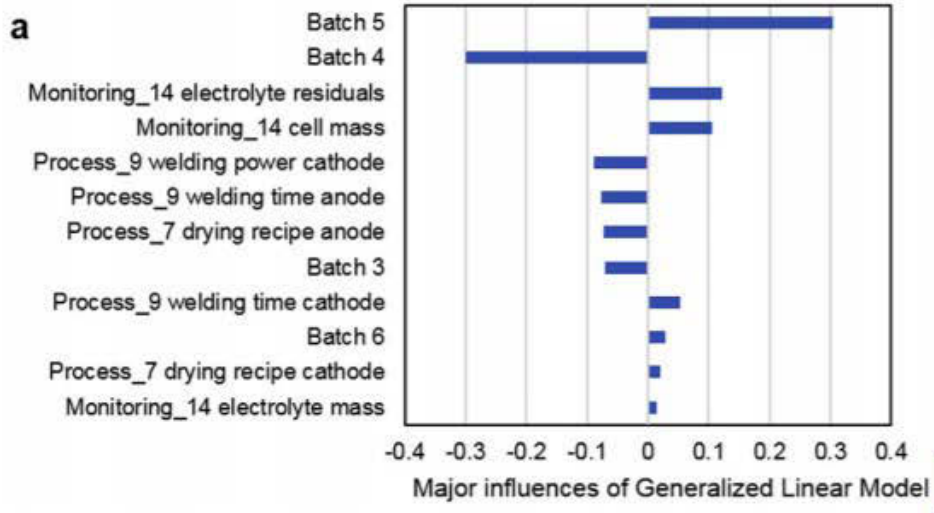



**Table 1**
Comparison of data mining tools for prediction of cell capacity before pre-charge and formation. The data set consists of 113 cells and 64 independent variables with an average capacity of 3.94 Ah.

| Cell capacity | Mean | GLM | ANN | SVR | DT | RF | GBT |
|---|---|---|---|---|---|---|---|
| RMSE (in Ah) | 0.89 | 0.42 | 0.76 | 0.48 | 0.51 | 0.44 | 0.42 |
| Mean relative error (in %) | | 7.6 | 15 | 9.1 | 7.8 | 7.8 | 6.6 |
| Improvement compared to mean (in %) | | 52 | 14 | 46 | 43 | 50 | 52 |

**Generalized Linear Model (continuous)**

**Decision Tree (discrete)**



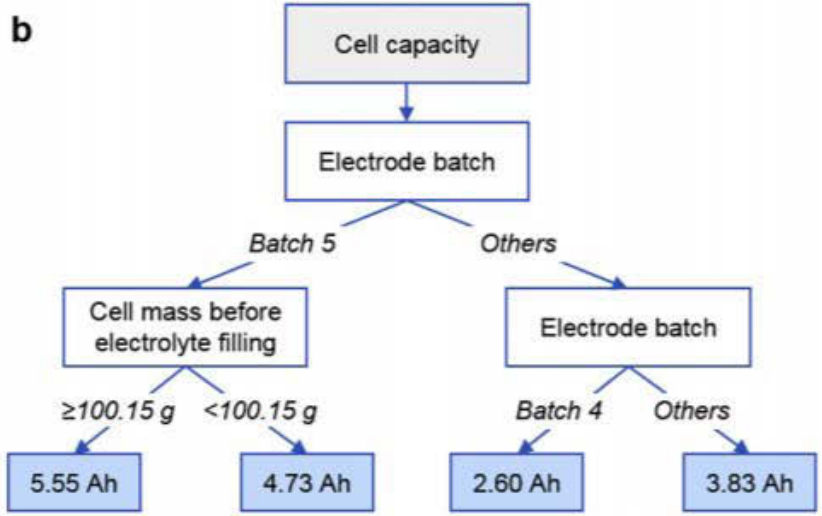




**Fig. 5.** Influences on cell capacity as derived from the GLM model (image a) and the DT model (image b).

# PART 2

Penalized Regression, Cross Validation

Early Life Prediction Example

# Part 2 – Penalized Regression & Cross Validation

- Linear least squares
- Penalized Regression
  - Ridge
  - LASSO
  - Elastic Net
  - Logistic regression → Neural Net.
- Determining hyperparameters via cross-validation

- Battery Example
  - Early life prediction
  - Fast charge protocol optimization

Goals

- Reveal complex relationships that are unknown a priori
- Down-select features/mechanisms that best describe data
- Avoid under/overfitting
- Determine when more test data are needed

- Reduce the time duration of battery lifetime experiments

# Penalized Regression

….(mostly) following notation of Andrew Ng

# Cost function (linear least squares)

- Hypothesis function: $h = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \ + \theta_n x_n$

$$h_\theta(x) = [\theta_0 \quad \theta_1 \quad \cdots \quad \theta_n] \begin{bmatrix} x_0 = 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta X$$

  - $n^{th}$ order hypothesis function or model
  - $m$ data observations
- Cost function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- Minimization objective – $\min_\theta J(\theta)$

- Parameters
- Independent variables or "**features**"
- Can include nonlinear features, e.g.
  - $x_1 = z$
  - $x_2 = z^2$
  - $x_3 = z^3$
- Dependent variable, observation

# Least squares solution - Analytical

- Analytical solution of linear problem for parameters:

$$\theta = (X^T X)^{-1} X^T y$$

- Matlab solution: theta = pinv(X'*X)*X'*y
- Inverse not possible if $(X^T X)$ is singular
  - redundant/linearly dependent features
  - too many features

… delete some features (or use regularization!)

# Least squares solution – Generic numerical algorithm

- Gradient descent – similar to Newton's method for solving nonlinear equations
  - Make initial guess, $\theta_j$
  - Iterate {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

    }
  - learning rate, $\alpha$
  - prevents oscillation - plot cost function for each iteration
- More advanced numerical algorithms (Matlab):
  - Unconstrained: fminunc, fminsearch
  - Constrained: fminbnd, fmincon

# Scaling Data – Improves Convergence

- Goal: Get all variables to vary between -1 to +1 or similar (doesn't need to be exact)
- Normalize with maximum range or standard deviation of each variable

Example model and data

$$q = 1 - at^{\frac{1}{2}} - bN^2$$

$$h = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$t \in$ (0 to 365) , $N \in$ (0 to 10,000), $q \in$ (0.6 to 1.0)

$t^{\frac{1}{2}} \in$ (0 to 19), $N^2 \in$ (0 to 100,000,000)

$q$ : capacity (relative)
$t$ : time (days)
$N$ : cycles

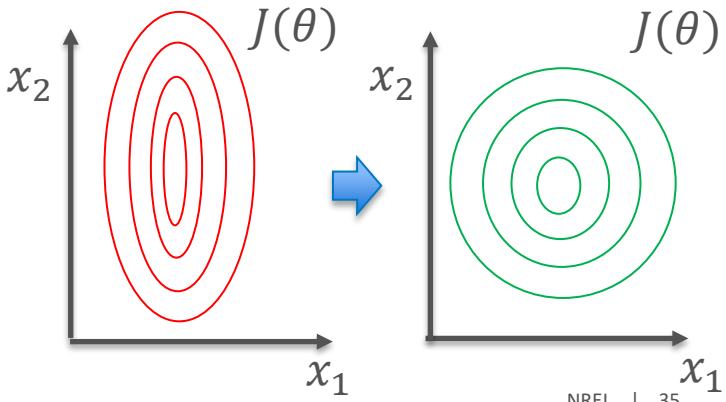
Scaled using max range

$$x_1 = \frac{t^{\frac{1}{2}}}{\max\left(t^{\frac{1}{2}}\right)}, \qquad x_2 = \frac{N^2}{\max(N^2)}$$

$$y = \frac{q - \min(q)}{\max(q) - \min(q)}$$

Centered & scaled using standard deviation

$$y = \frac{q - \text{mean}(q)}{\text{std}(q)}$$



# Weighting data

- Introduce weights, $w_i$, into cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m} w_i(h_\theta(x^{(i)}) - y^{(i)})^2$$

- Ideal weight is the reciprocal of the variance of the error
- Observations with small variances should have relatively large weights. Observations with large variances should have relatively small weights
- "Robust" regression de-weights outlier data points
- Use your intuition: Are some data points more important than others?

# Regularization – helps solve ill-posed problems

- Cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

- Analytical solution of linear problem:

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

- Large $\lambda$ forces parameters to be smaller. Helps convergence. Helps prevent overfitting
- How to choose hyperparameter, $\lambda$
  - Sweep range of values. Check results versus cross-validation data

# Cross validation – Train & Test!

Training data

Testing data

Goal: Avoid under- and over-fitting


Figure: https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html

# k-fold Cross Validation


**Total Error**
for model hypothesis
with hyperparameter value $\lambda$

- Divide data sets into “k” random bins
  - “Hold one out” removes one at a time (identical if k = # total observations)
- Provides method to optimize hyperparameter and validate model hypotheses

Figure: http://www.ebc.cat/2017/01/31/cross-validation-strategies/#k-fold

# Selecting Best Model Using Cross Validation

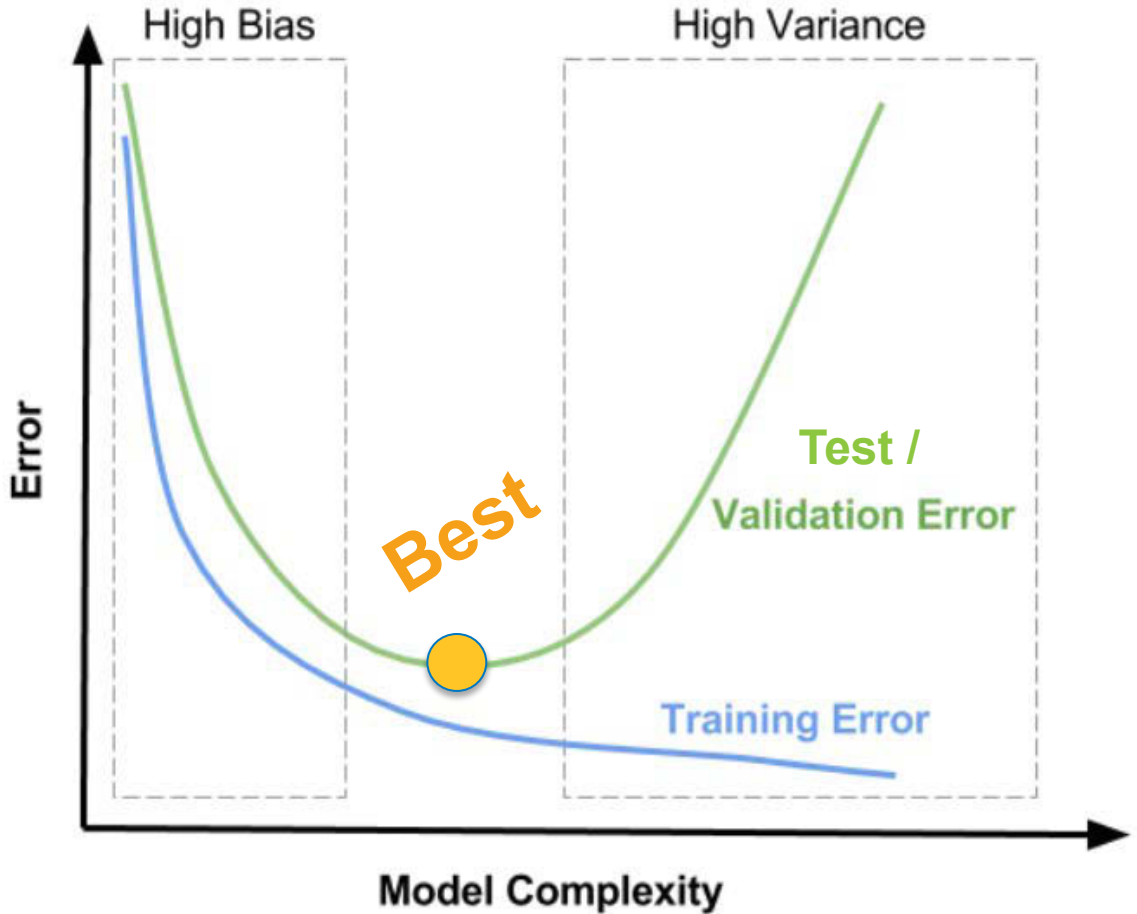



Figure: http://www.luigifreda.com/2017/03/22/bias-variance-tradeoff/

- Also useful to plot convergence of model versus amount of test data
- For high variance
  - More test data
  - More regularization (less model complexity)
  - Better features

# LASSO (least absolute shrinkage and selection operator)

- Cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}|\theta_j|$$

- Hyperparameter, $\lambda$
- **Large $\lambda$ forces parameters to be zero** if their contributions are insignificant (throws out model terms).
- The $L_1$ norm is non-differentiable. No analytical solution exists.
- Now your algorithm can make decisions to select relevant features (e.g. mechanisms) for a model!!

# Elastic Net = LASSO + Regularization

- Cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda_1 \sum_{j=1}^{n}|\theta_j| + \lambda_2 \sum_{j=1}^{n}\theta_j^2$$

- Contains both
  - $L_1$ norm = Manhattan or taxicab norm

$$= \sum_{j=1}^{n}|\theta_j| = \|\theta\|_1 \quad \text{(LASSO)}$$

  - $L_2$ norm = Euclidian norm

$$= \sum_{j=1}^{n}\theta_j{}^2 = \|\theta\|_2 \quad \text{(Regularization)}$$

# Logistic Regression – Classification, e.g. y={0,1}

- Recall our linear regression hypothesis function

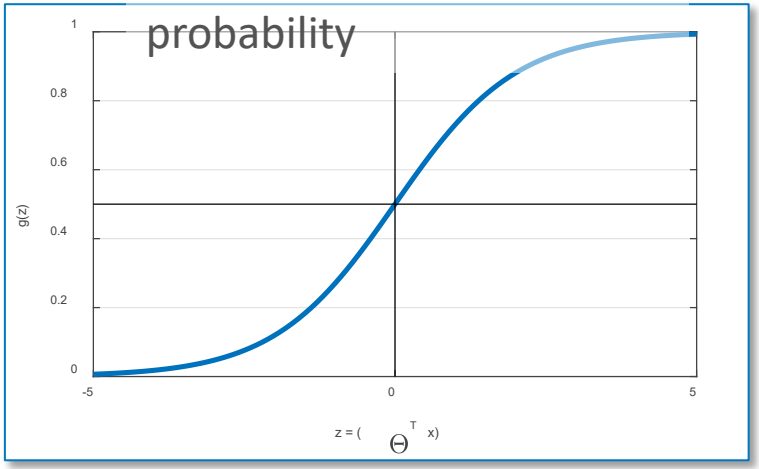
$$h_\theta(x) = \theta^T x$$

- Logistic regression uses a non-linear hypothesis function that switches on/off → classification, y = {0, 1} or {0, 1, 2, …}

$$h_\theta(x) = g(\theta^T x) \quad \text{where} \quad g(z) = \frac{1}{1+e^{-z}}$$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + (1-y^{(i)}) log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

Output of logistic function also can be interpreted as probability




- …extends to Neural Network

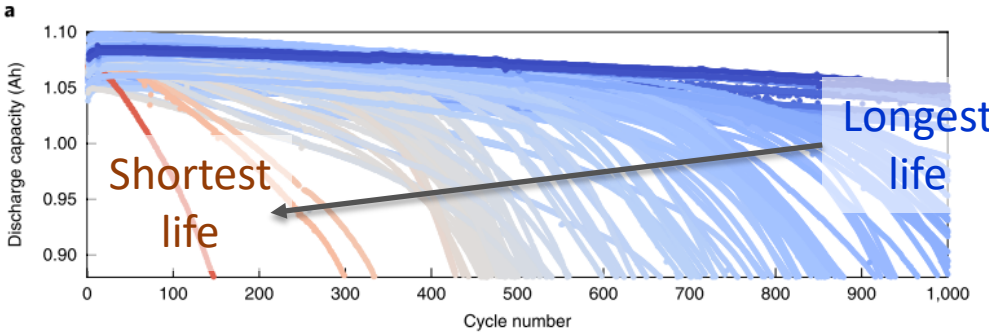
$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(h_\Theta\left(x^{(i)}\right)\right)_k + (1-y_k^{(i)}) log\left(1 - \left(h_\Theta\left(x^{(i)}\right)\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(\Theta_{ji}^{(l)}\right)^2$$

observations — outputs — layers — neurons in layer $l$

# Early Lifetime Prediction using Penalized Regression

# Early Life Prediction

*Severson et. al., Nature Energy 4 (5) 2019*



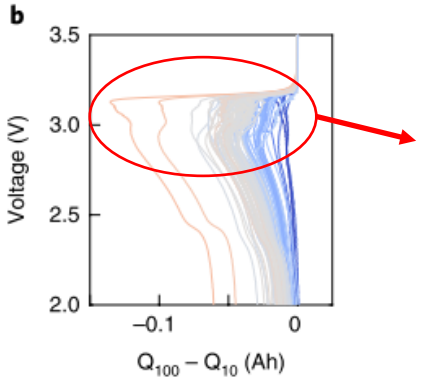

"Data-driven prediction of battery cycle life before capacity degradation"

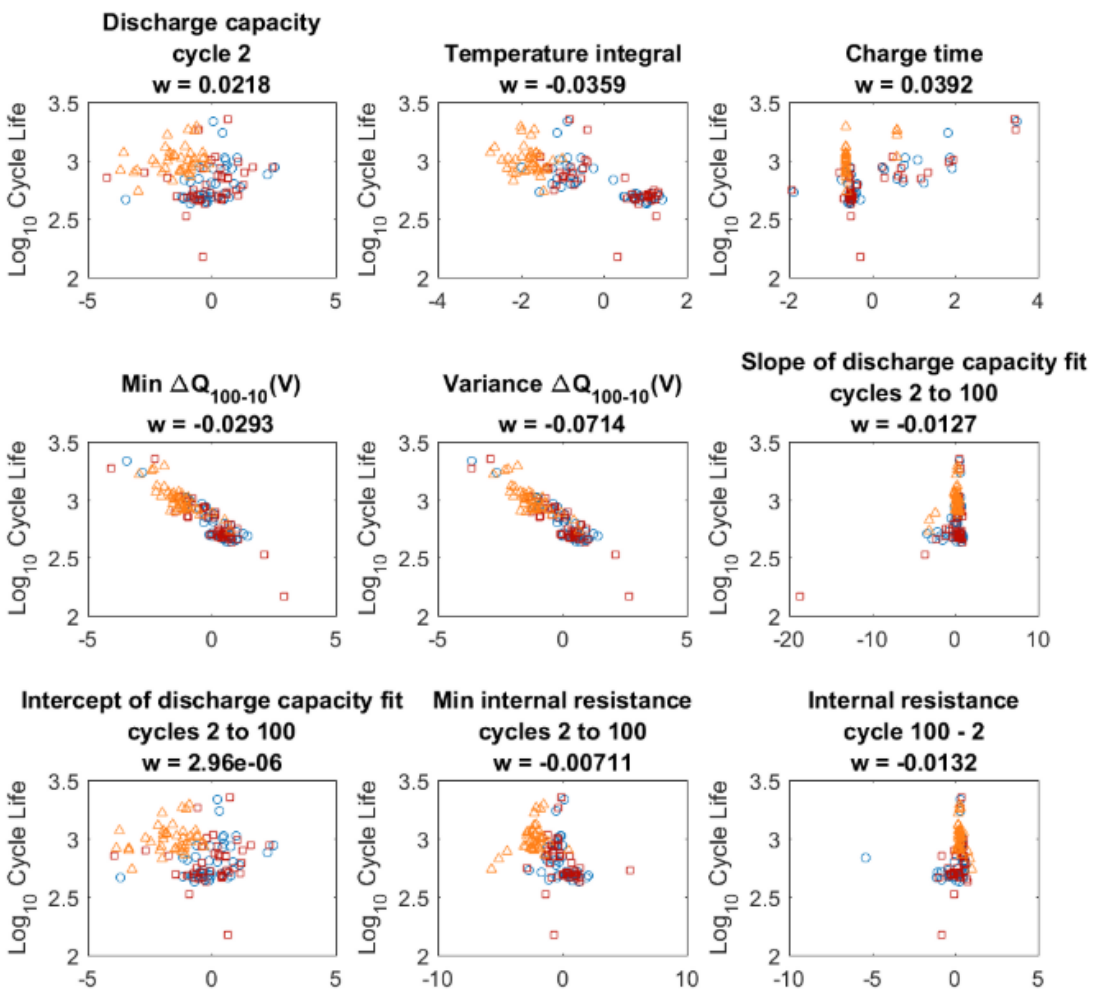
- 124 cells (LFP/Gr by A123, 1.1 Ah, 2.0V-3.6V, 4C CC-CV discharge*)
- Various charging protocols:
  - 3.6C to 7C CC charge in different combinations of steps from 0%-80% SOC, 1C CC-CV from 80%-100% SOC for all cells*
- Machine learning models:
  - Regression: linearized model regularized by both **lasso** and **elastic net**, with hyperparameters optimized by **four-fold cross validation** and Monte Carlo sampling
  - Classification: **logistic regression**
  - Code provided open source

*All cells share the same discharge voltage curves

# Elastic Net Considered Many Features

*Severson, Nature Energy 4 (5) 2019*



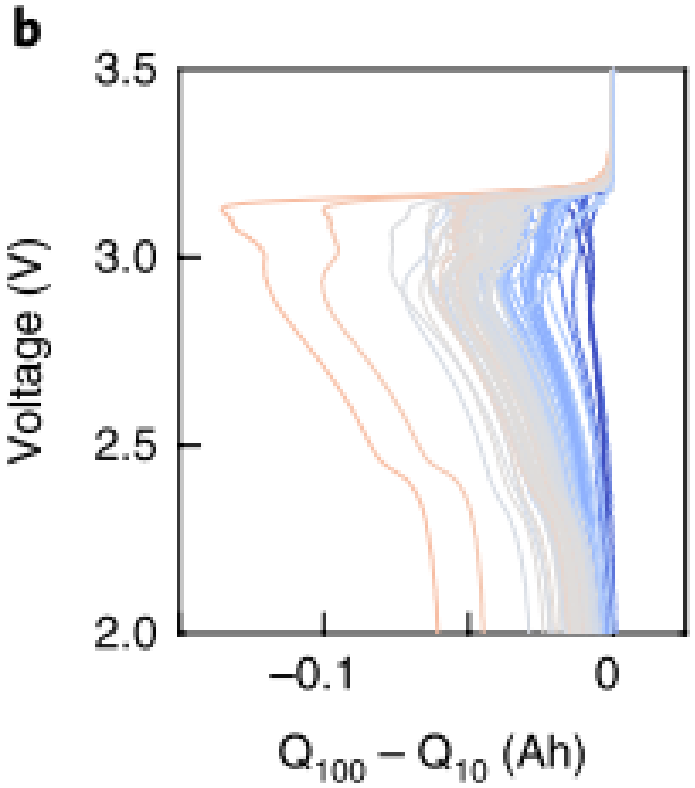




- Best is difference between discharge curve at cycle 100 and discharge curve at cycle 10

# Capacity Difference Metric

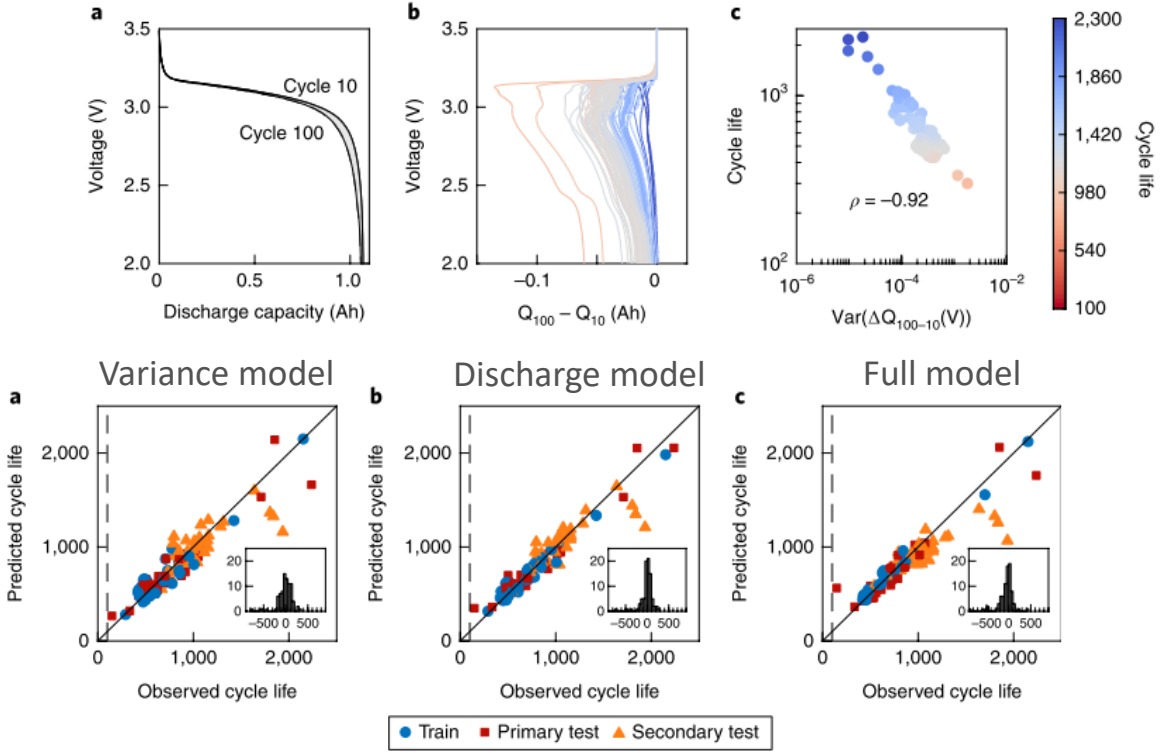
*Severson, Nature Energy 4 (5) 2019*

- $Q_{100} - Q_{10}$ = ΔCapacity at a given V
  - Discretized voltage with 1000 values from 2.0 to 3.6V
  - Resampled capacity data using splines to obtain 1000 values of capacity, $Q_i$, for each cycle I
- Enables comparison of two different discharge curves that have a different number of time stamps



# Unique descriptor and model results

*Severson, Nature Energy 4 (5) 2019*




**Variance model:**
Uses only var($Q_{100}$-$Q_{10}$) as a descriptor

**Discharge model:**
Uses only discharge curve information as descriptors (6 total)

**Full model:**
Discharge model + charge time, temperature, and resistance descriptors (9 total)

[1]

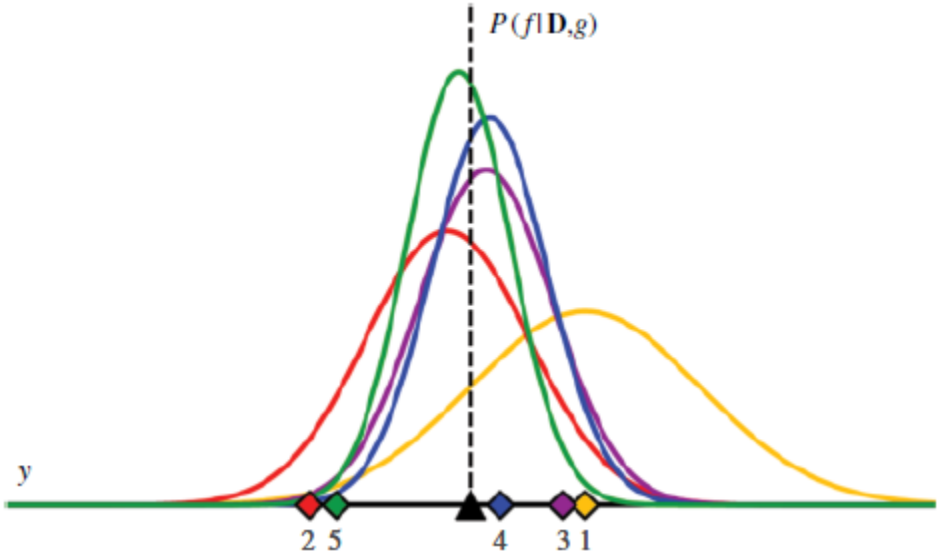
# Discussion

*Severson, Nature Energy 4 (5) 2019*

- Result is similar to incremental capacity analysis (dQ/dV)
  - dQ/dV generally run at slow rates << C/5
  - 4C discharge includes impedance growth effects
- LFP: Single capacity fade mode (Li loss)
- Full discharge curve available from every cell, every cycle
  - Highly valuable model provided this discharge condition always exists
- Can this be applied for SOH estimation in real-world use?
  - End of charge is a more common operating condition (PART3, GPR examples)
- Accelerates laboratory testing (PART3, Bayesian Opt. of Charge Protocols)

# Bayesian Optimization Concept

- Posterior distribution changes (prediction improves) as more training data added
- Black triangle is truth, f
- Diamonds are training data, D
  - First
  - Second
  - Third
  - Fourth
  - Fifth

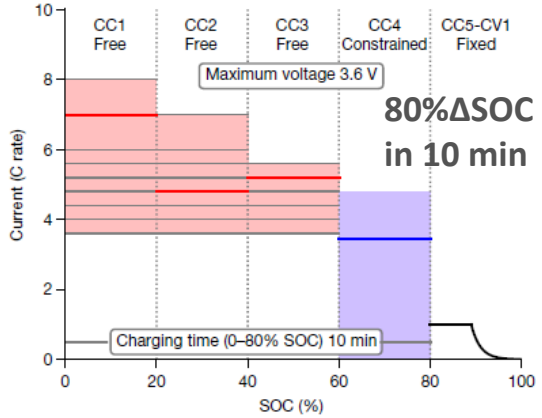



Source: T. Mueller, A.G. Kusne, R. Ramprasad, Reviews in Computational Chemistry (2016)

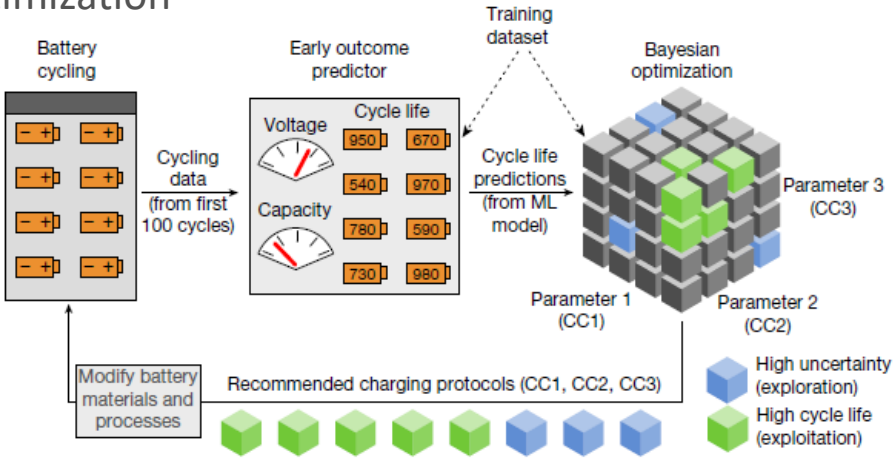
# Fast-charge protocol optimiz.

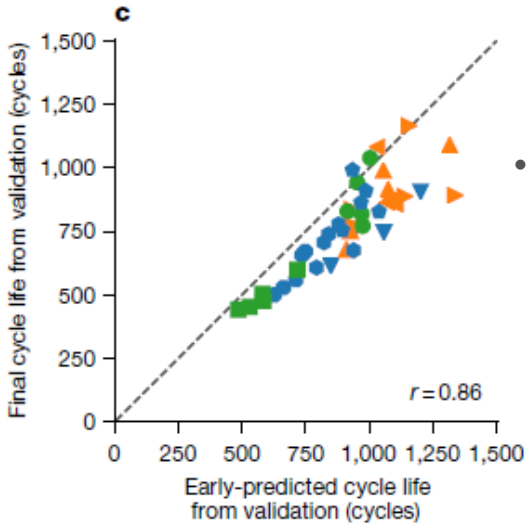
(P. Attia, Nature, 2020)

- Find charge protocol (CC1,2,3,4) with longest life

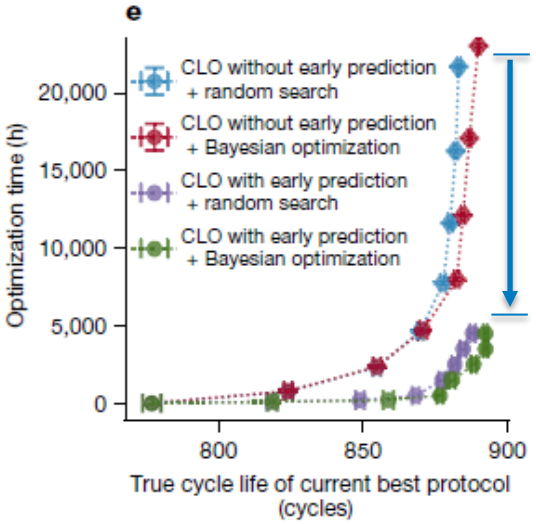



- Closed-loop experiment optimization




- Uses early-life prediction method of Attia, 2019.




**Early life prediction accelerates optimization**

**Bayesian optimization**

# Conclusions & Related Work

- Penalized regression + cross validation automates selection of most important features when fitting a regression model
  - Features should embed physical knowledge
    - Physical properties, mechanisms, model calcs., similarity transforms, etc.
  - Avoids under/over fitting
- Related – Hierarchical ML[3-6]
  - Leverages domain knowledge to reduce # experiments required to explore large variable spaces (enable ML on small datasets)
  - Guides the choice of descriptors that are explicitly dependent on material properties, known degradation mechanisms, etc.
- Present work in Hierarchical ML at NREL & INL[7]
  - Identify physical life models describing lifetime across variable operating conditions
  - Reduce testing, increase accuracy, automate

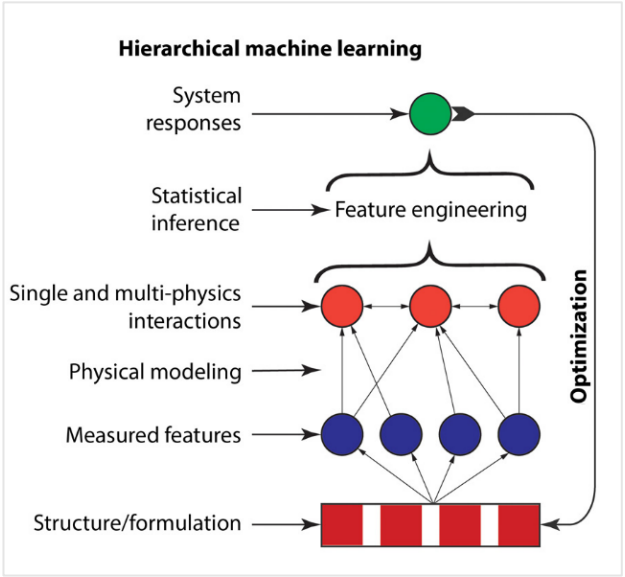



Figure: Childs MRS Comm. (2019)

# References


[1] K. A. Severson *et al.*, “Data-driven prediction of battery cycle life before capacity degradation,” *Nat. Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[2] P.M. Attia *et al.*, “Closed-loop optimization of fast-charging protocols for batteries with machine learning, *Nature*, vol. 578, p. 397, 2020.

[3] C. M. Childs and N. R. Washburn, “Embedding domain knowledge for machine learning of complex material systems,” *MRS Commun.*, no. 3, pp. 806–820, 2019.

[4] A. Menon *et al.*, “Elucidating multi-physics interactions in suspensions for the design of polymeric dispersants: A hierarchical machine learning approach,” *Mol. Syst. Des. Eng.*, vol. 2, no. 3, pp. 263–273, 2017.

[5] A. Menon, J. A. Thompson-Colón, and N. R. Washburn, “Hierarchical machine learning model for mechanical property predictions of polyurethane elastomers from small datasets,” *Front. Mater.*, vol. 6, no. May, pp. 1–12, 2019.

[6] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, “Big data of materials science: Critical role of the descriptor,” *Phys. Rev. Lett.*, vol. 114, no. 10, pp. 1–5, 2015.

[7] P. Gasper, “Hierarchical Machine Learning For Identification of More Accurate Battery Calendar Lifetime Models,” *in preparation.*

# PART 3 – Probabilistic Methods

Bayes Theorem

- Bayesian Optimization Fast Charge Example

Gaussian Process Regression

- Lifetime/Health Examples

# Bayes Theorem

- Example: Cancer test with 90% accuracy, $p(\text{test+} \mid \text{cancer})$
  - Probability of cancer given positive test, $p(\text{cancer} \mid \text{test+})$

What you know

What you want to know

- Start with **prior** probability information.
  - Use new information to calculate **posterior** probability

- **Bayes Theorem** lets us reverse conditional probabilities

*Conditional probability for event A given event B* x *Marginal probability for event B*

*Conditional probability for event B given event A*

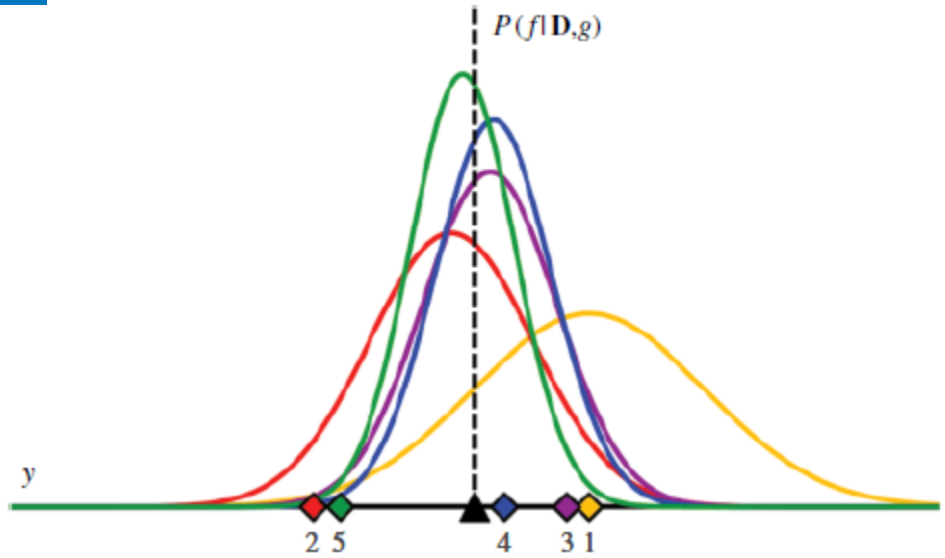
$$p(B|A) = \frac{p(A|B)p(B)}{p(A)}.$$

÷ *Marginal probability for event A*

Source: http://www.stat.cmu.edu/~brian/463-663/week09/Chapter%2003.pdf

# Bayesian Optimization Concept

- Posterior distribution changes (prediction improves) as more training data added
- Black triangle is truth, f
- Diamonds are training data, D
  - First
  - Second
  - Third
  - Fourth
  - Fifth

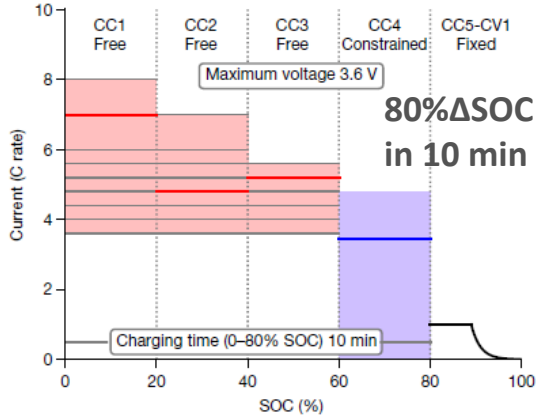



Source: T. Mueller, A.G. Kusne, R. Ramprasad, Reviews in Computational Chemistry (2016)

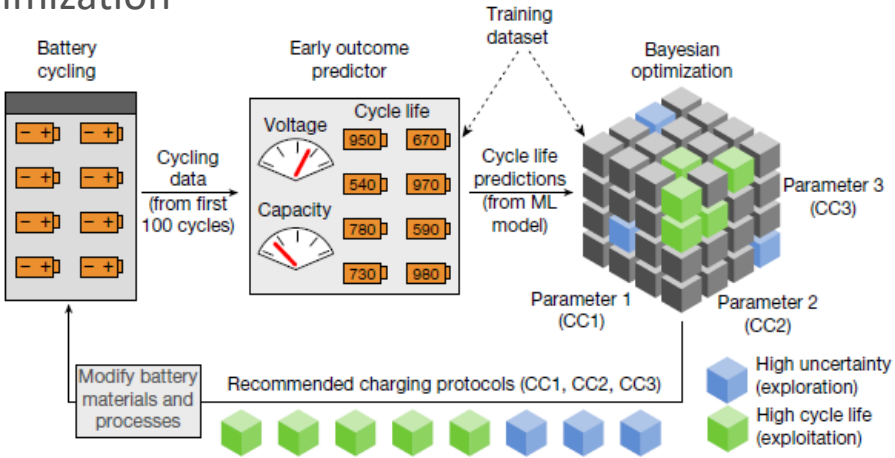
# Fast-charge protocol optimiz.

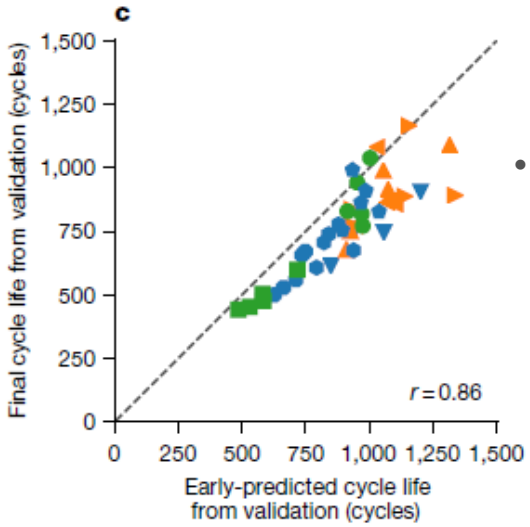
(P. Attia, Nature, 2020)

- Find charge protocol (CC1,2,3,4) with longest life

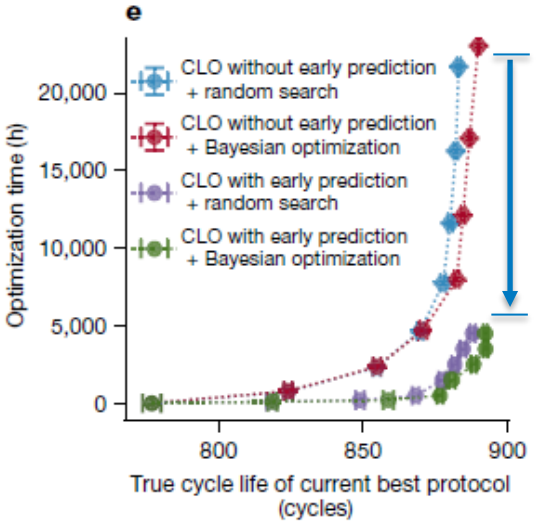



- Closed-loop experiment optimization




- Uses early-life prediction method of Attia, 2019.




**Early life prediction accelerates optimization**

**Bayesian optimization**

# Gaussian Process Regression (GPR) and State-of-Health (SOH) Estimation

- Motivation
- GPR Algorithm
- Examples
- Conclusions, Future Work, Open-source Data

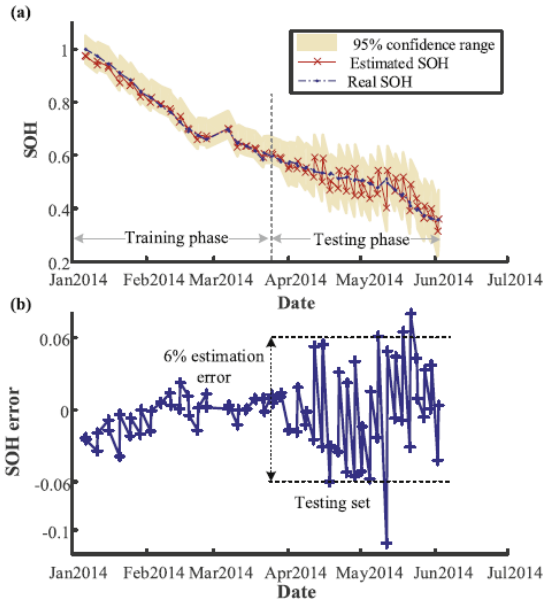
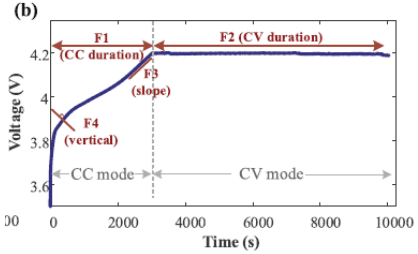



Fig. 7. SOH estimation results of battery RW9 with random walk operation.

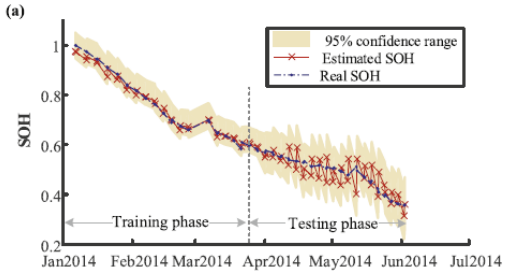
*Figure: D. Yang, JPS, 2018*

# Motivation – Gaussian Process Regression (GPR) Algorithm

**Motivation** – Estimate full capacity (SOH) without running a full charge/discharge




Features
1. Time in CC mode
2. Time in CV mode
3. Voltage relaxation
4. Final voltage
5. EIS spectra




## GPR benefits

- Ability to down-select best "features" that correlate with capacity
- Once trained, easy to implement in real-time controller
- Provides capacity "measurement" and measurement error
- Interpretable, can include physical knowledge

# Gaussian Process Regression (GPR) Algorithm

# Gaussian Process Regression

- Probability distribution between measured input **x** (observation) and output **y** (response)

Noise covariance matrix

"Prior" distribution $\mathbf{y} \sim N(0, \mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}_n)$

Distance between $x_i$ and $x_j$. When small, k is large.

Kernal $\mathbf{K}_f(\mathbf{x}, \mathbf{x}) = k_{ij} = \sigma_f^2 \exp\left(\frac{-(x_i - x_j)^2}{2l^2}\right)$

- Squared-exponential covariance function
- Many others possible

Measurement is important when weight $l$ is large

- Training

Optimize hyperparameters $\Theta = [\sigma_f, l, \sigma_n]$

Source: Yang, J. Power Sources, 2018

# Gaussian Process Regression - Optimization

- Training

  Optimize hyperparameters $\Theta = [\sigma_f, l, \sigma_n]$

- Cost function

$$L = \log p(\mathbf{y}|\mathbf{x}, \Theta) = -\frac{1}{2}\log(\det(\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n)) - \frac{1}{2}\mathbf{y}^T[\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n]^{-1}\mathbf{y} - \frac{n}{2}\log 2\pi$$

- Partial derivative for gradient descent algorithm

$$\frac{\partial}{\partial\Theta_i}\log p(\mathbf{y}|\mathbf{x}, \Theta) = \frac{1}{2}tr\left\{\left[\alpha\alpha^T - (\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n)^{-1}\frac{\partial(\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n)}{\partial\Theta_i}\right]\right. \qquad \alpha = [\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n]^{-1}\mathbf{y}$$

Source: Yang, J. Power Sources, 2018

# Gaussian Process Regression - Implementation

- Prediction

  "Posterior" distribution $p(\mathbf{y}^*|\mathbf{x}, \mathbf{y}, \mathbf{x}^*)$

- Estimate

$$\bar{\mathbf{y}}^* = K_f(\mathbf{x}, \mathbf{x}^*)^T[\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n]^{-1}\mathbf{y}$$

- Error estimate

$$\text{cov}(\mathbf{y}^*) = \mathbf{K}_f(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_f(\mathbf{x}, \mathbf{x}^*)^T[\mathbf{K}_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}_n]^{-1}\mathbf{K}_f(\mathbf{x}, \mathbf{x}^*)$$

$$95\%\ confidence\ range = \overline{SOH}_{estimate} \pm 1.96 \times \text{cov}(SOH_{estimate})$$

Source: Yang, J. Power Sources, 2018

# Examples of SOH Estimation Using GPR

# GPR – SOH Using Charge Features

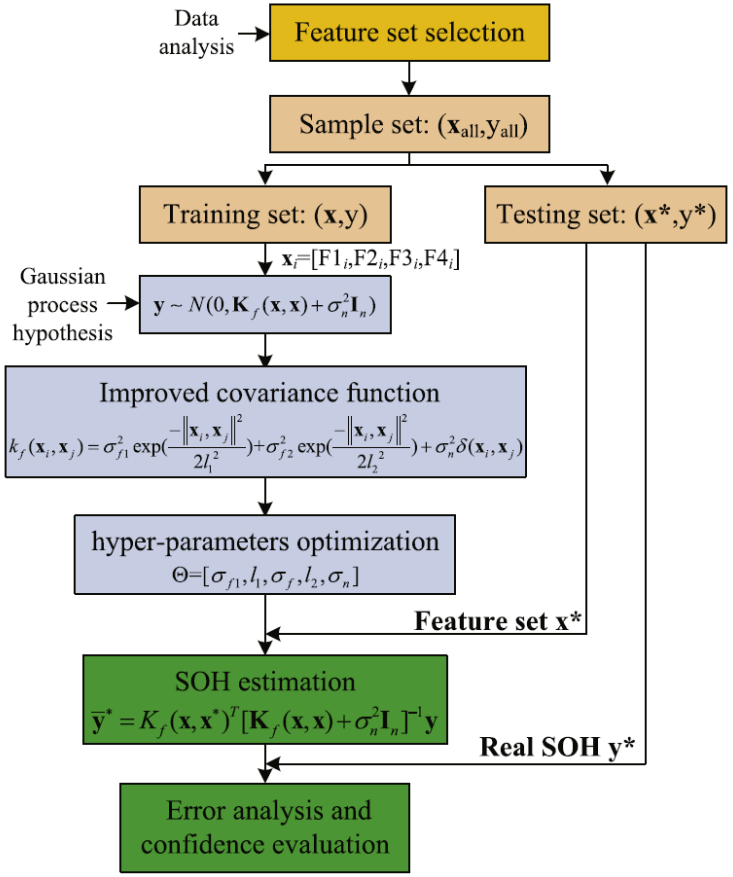



Fig. 4. The SOH estimation procedure.


ELSEVIER

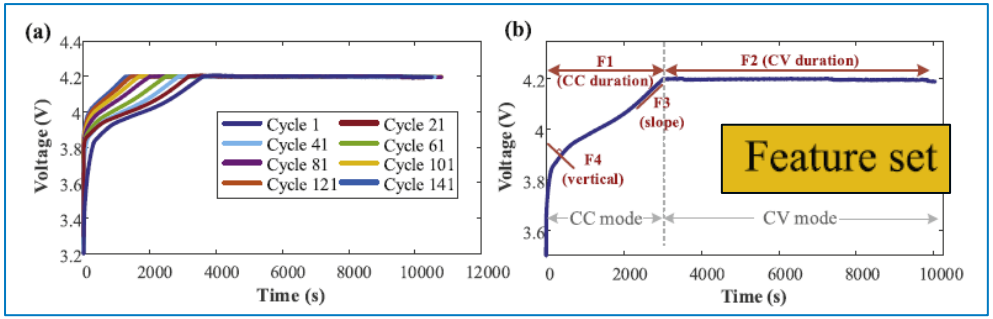
Journal of Power Sources

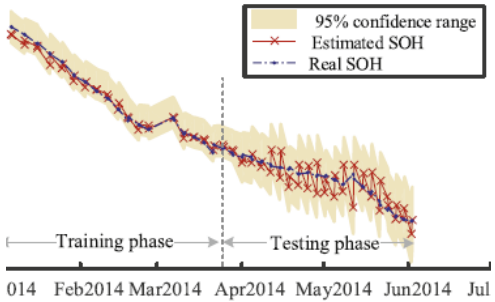
journal homepage: www.elsevier.com/locate/jpowsour

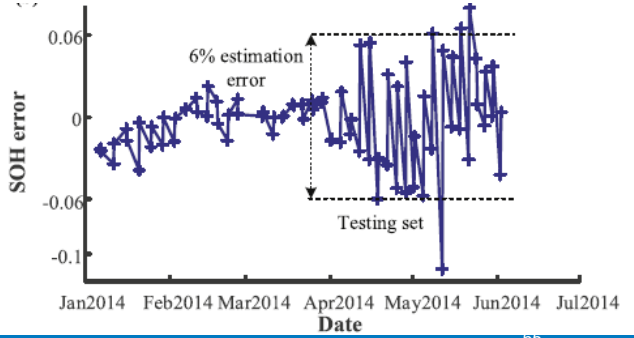

## A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve


Duo Yang, Xu Zhang, Rui Pan, Yujie Wang, Zonghai Chen*

*Department of Automation, University of Science and Technology of China, Hefei 230027, PR China*

# GPR SOH based on CCCV features ($time_{CC}$, $time_{CV}$)

Contents lists available at ScienceDirect

Engineering Failure Analysi

journal homepage: www.elsevier.com/locate/e

## Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: A case on lithium-ion batteries used in electric vehicles

Jing Feng [a,*], Paul Kvam [b], Yanzhen Tang [a]

[a] National University of Defense Technology, Changsha, HN, China
[b] Georgia Institute of Technology, Atlanta, GA, USA

ARTICLE INFO

*Article history:*
Received 25 August 2015
Received in revised form 8 April 2016
Accepted 12 April 2016
Available online 25 May 2016

*Keywords:*
Remaining useful lifetime
Marker processes
Bivariate Wiener process
Lithium-ion battery
Electric vehicle

ABSTRACT

Remaining useful lifetime (RUL) refers to the available service time left before the performance of a system degrades to an unacceptable level. Recent innovations to lithium-ion battery packs have raised expectations with regard to energy storage capability in electric vehicles (EVs). This has catalyzed new research on RUL prediction, since accurate RUL prediction for lithium-ion batteries used in EV is highly desired for safe and lifetime-optimized operation. A battery's maximum releasable capacity (MRC) usually decays over time, thus it is a primary factor which determines the remaining cycle life of the battery. However, MRC usually needs to be measured under strict laboratory conditions and cannot be easily assessed during field use in EVs. This naturally inhibits potential applications of many online RUL prediction methods that rely on MRC measurements. We found two markers of MRC decay, named as time-to-voltage-saturation (TVS) and time-to-current-saturation (TCS), from constant-current constant-voltage charging (CC/CV) curves, which can be used in place of MRC measurements during field use. We propose a RUL prediction method based on a damage-marker bivariate degradation model in which one term represents damage (MRC decay), the other represents a composite marker constructed from TVS and TCS. We model this degradation process using a two-dimensional Wiener process to obtain the RUL distribution, using method of maximum likelihood for population parameters' estimation. Bayesian methods are used to update the estimators of parameters with online data. The effectiveness of the model is verified with public data of four 18,650 batteries from NASA.



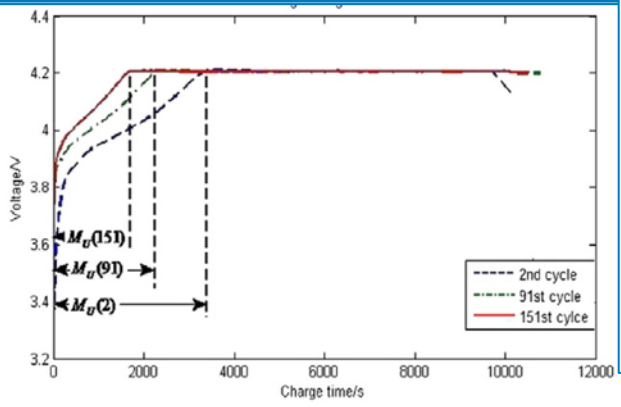



**Fig. 3.** Extraction of time-to-voltage-saturation (TVS) from CC/CV charge voltage curve.

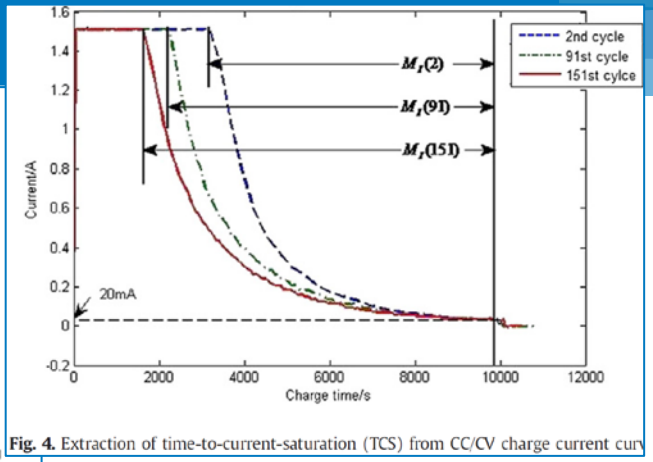



**Fig. 4.** Extraction of time-to-current-saturation (TCS) from CC/CV charge current curv

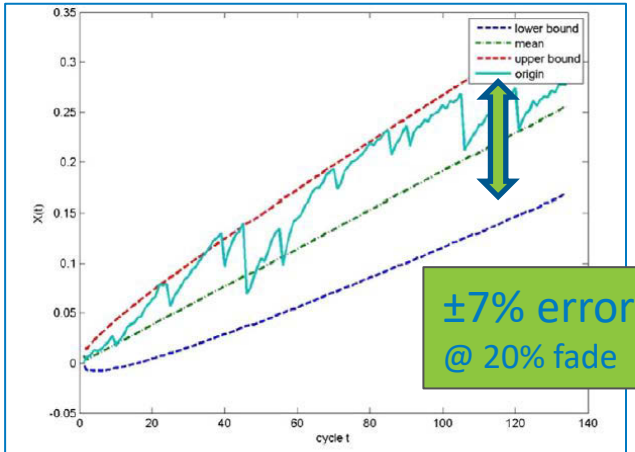



**Fig. 13.** Observed values along with point estimates and 95% confidence intervals for X(t).

# GPR SOH using relaxation features (Vs, Ro, Ra, Rc)

Electrochimica Acta

journal homepage: www.elsevier.com/locate/electacta

## State-of-health (SOH) evaluation on lithium-ion battery by simulating the voltage relaxation curves

Kun Qian [a, c], Binhua Huang [b, c], Aihua Ran [a, c], Yan-Bing He [b], Baohua Li [b, *], Feiyu Kang [a, b, c, **]

[a] Shenzhen Environmental Science and New Energy Technology Engineering Laboratory, Tsinghua-Berkeley Shenzhen Institute, Shenzhen, 518055, PR China
[b] Engineering Laboratory for the Next Generation Power and Energy Storage Batteries, Graduate School at Shenzhen, Tsinghua University, Shenzhen, 518055, PR China
[c] Laboratory of Advanced Materials, School of Materials Science and Engineering, Tsinghua University, Beijing, 100084, PR China

ARTICLE INFO

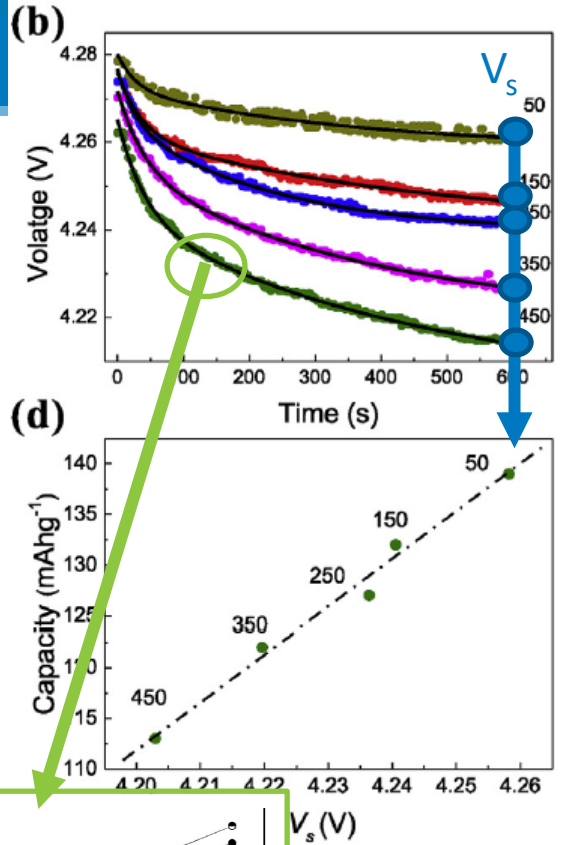
*Article history:*
Received 30 November 2018
Received in revised form
21 January 2019
Accepted 12 February 2019
Available online 13 February 2019

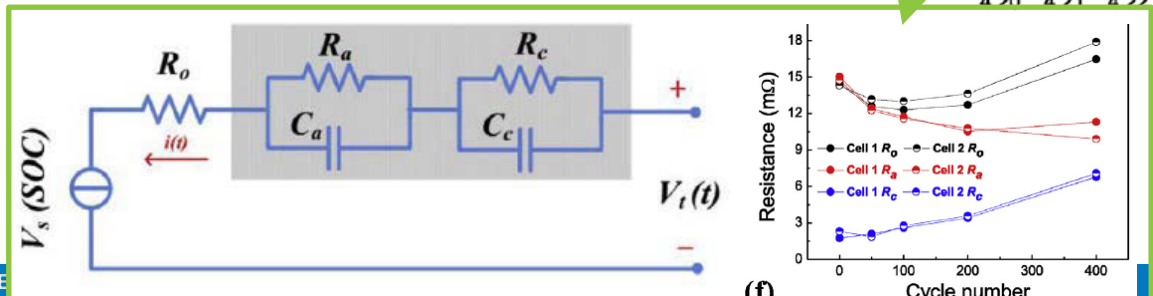
*Keywords:*
Lithium ion battery
State of health
Voltage stabilization model
Aging mechanism

ABSTRACT

State-of-health evaluation of lithium ion batteries is one of the key challenges to construct efficient battery system with enhanced performance. In this paper, we demonstrate a facile evaluation method which is based on the voltage stabilization behaviour during relaxation. A second-order equivalent circuit is employed to describe the voltage relaxation. By simulating the voltage relaxation curves, the characteristic parameters that can distinguish the ohmic, activation and concentration depolarizations are acquired. The obtained parameters are sensitive to electrochemical states, which provide comprehensive evaluation to the state-of-health of batteries and aging mechanism analysis. As an example, the proposed method identifies two aging stages of $Li(Ni_{1/3}Co_{1/3}Mn_{1/3})O_2$/graphite pouch cells during accelerated aging. Namely, a loss of lithium inventory in the first stage of aging, and the loss of active material in the next stage. Postmodern analysis including X-ray diffraction and high-resolution transmission electron microscope well agrees with the voltage relaxation simulation. The proposed state-of-health evaluation method is non-destructive and accessible for on-board aging diagnosis, which is of great significance for developing better battery management system.





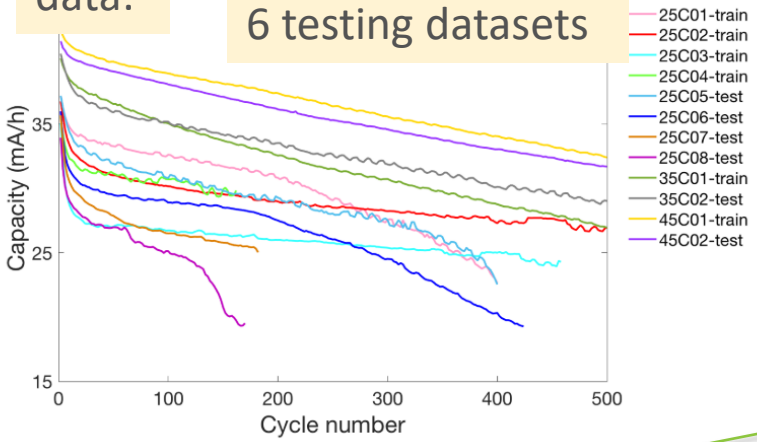



# GPR: SOH from EIS

(Zhang, Nat. Comm, 2020)

Aging data:

6 training datasets
6 testing datasets




- 9 EIS measurements were recorded every-other cycle.
  - 0.02 Hz - 20 kHz, 5mA excitation (C/9), 15-minute rest before

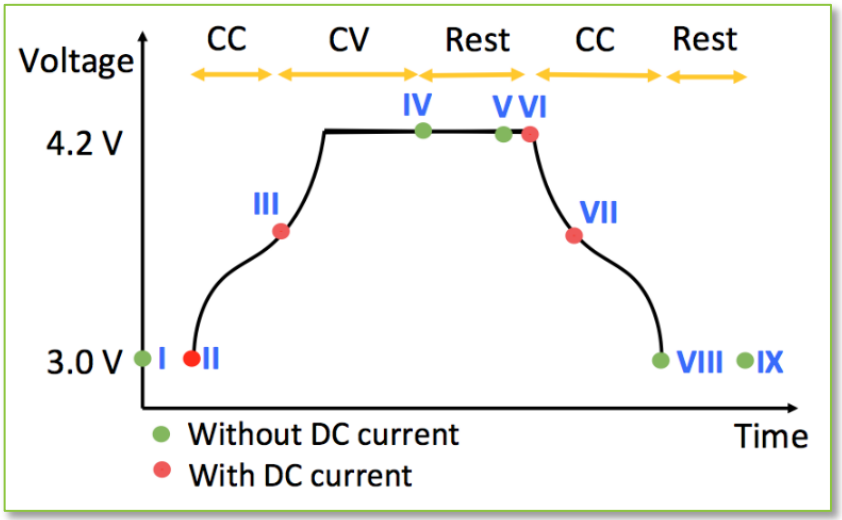
nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-15235-7 OPEN

Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning

Yunwei Zhang[1,2,6], Qiaochu Tang[2,3,4,6], Yao Zhang[5], Jiabin Wang[2,3,4], Ulrich Stimming[2,3,4,7] & Alpha A. Lee[1,2,7]



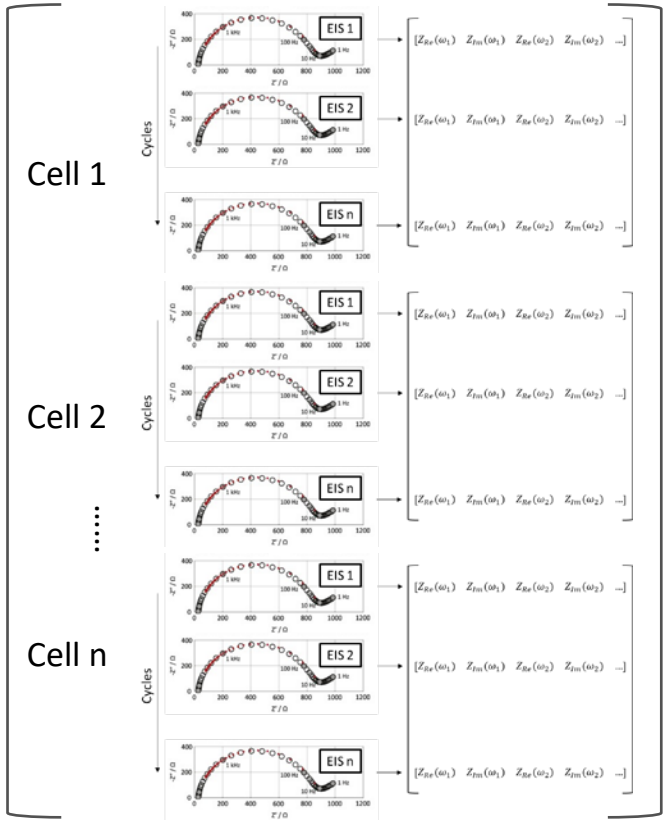
# GPR: SOH from EIS – Data structure

(Zhang, Nat. Comm, 2020)

**Input Matrix** *(N by M)*

(a.k.a. features or indep. variables)

$\mathbf{x} =$




**Response Vectors**

(a.k.a. output or dependent variables)

*(N by 1)*

$\mathbf{y_1} =$ Capacity at the end of the cycle where EIS was measured.

*(N by 1)*

$\mathbf{y_2} =$ Remaining useful life (# of cycles until relative capacity hits 80%)

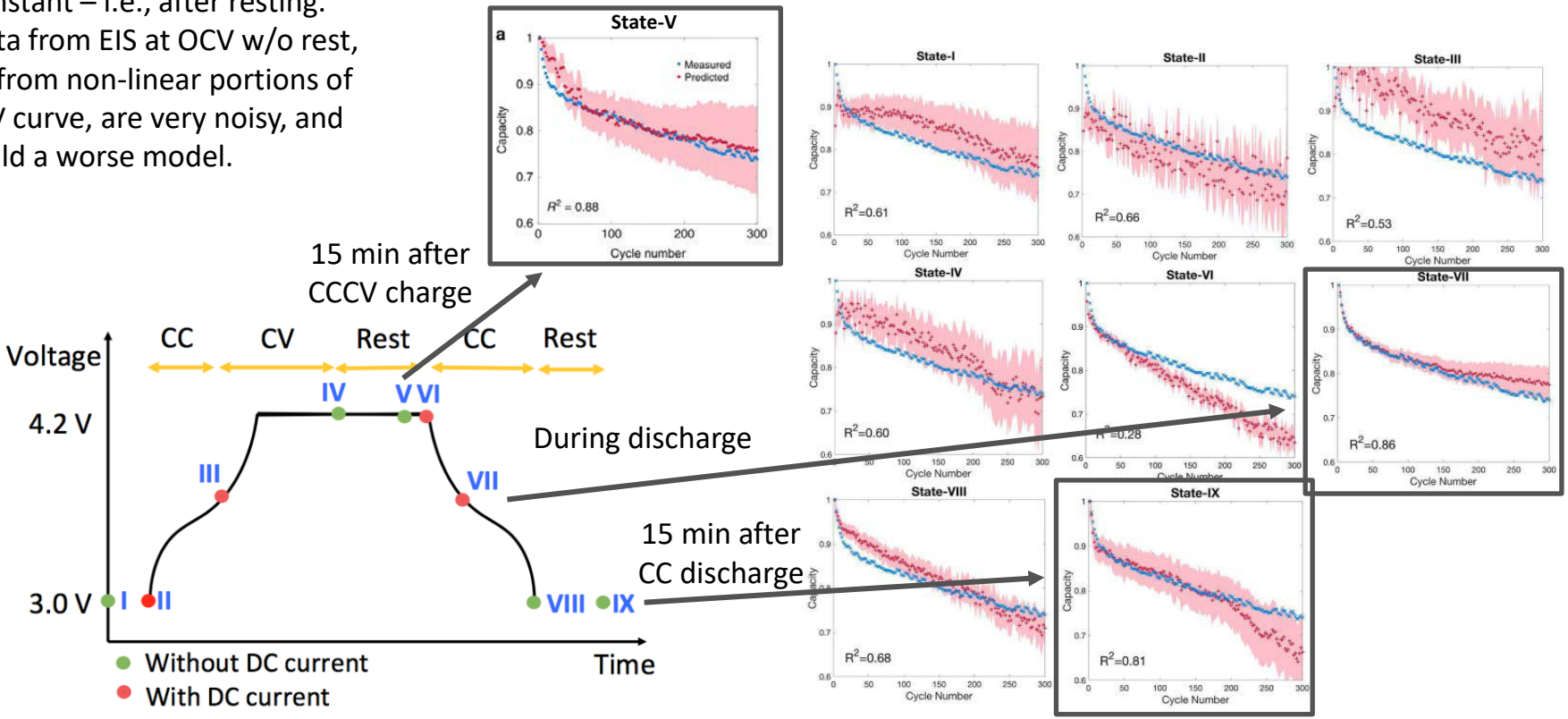
*N = total number of EIS measurements across all cells*

*M = number of data points per EIS measurement*

# GPR: SOH from EIS – Selecting Input Data

(Zhang, Nat. Comm, 2020)

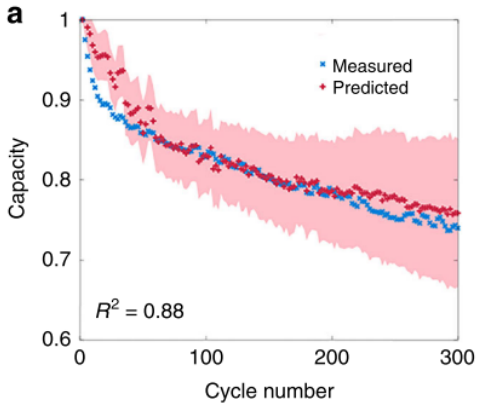
**Not all measurements are of equal value**. EIS must be conducted when battery state is constant – i.e., after resting. Data from EIS at OCV w/o rest, or from non-linear portions of C-V curve, are very noisy, and build a worse model.

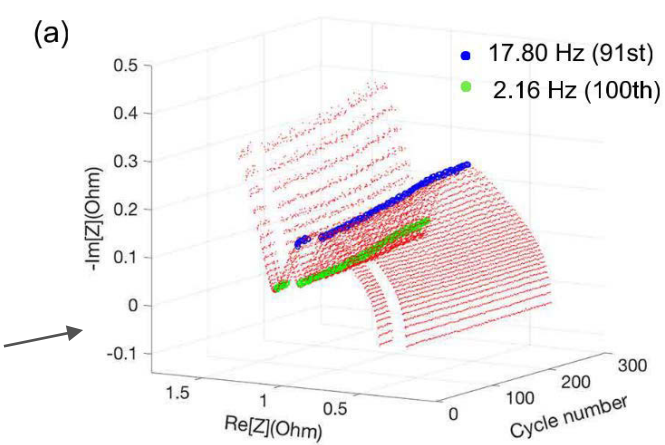


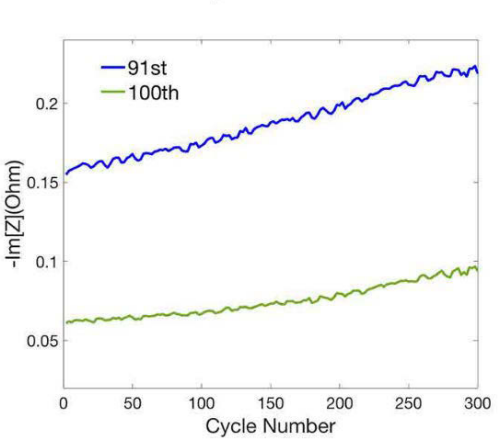
# GPR: SOH from EIS– Selecting Best Prediction Frequencies

(Zhang, Nat. Comm, 2020)

- GPR can be modified with hyperparameters that can automatically down-weight input features (i.e., specific frequencies from the EIS measurements) that are irrelevant to the regression
- This method essentially identifies the critical information from the input matrix that is required to make a prediction
  - Algorithm selected trough and peak of EIS semicircle



# Conclusions – Gaussian Process Regression for SOH

## Limitations of literature

- Algorithms tested on 2-12 cells
- Constant T
- Constant $SOC_0$ @ start of charging
- Constant charge rate

## → Must generalize for real-world operating conditions

- Variable T, $SOC_0$, charge rate
- Variable cycling/use/degradation modes
- Feature selection
  - Physics justified
  - Physics model parameters

## Features

1. Time in CC mode
2. Time in CV mode
3. Voltage relaxation
4. Final voltage
5. EIS spectra

→ Capacity in CC mode within a voltage window

*(Independent of charge rate, SOC)*

→ Model parameters $R_o$, $R_a$, $R_c$ fit to relaxation

*(Independent of charge rate)*

→ Model parameters $R_{ct}$, $R_w$, $C_{dl}$ fit to EIS

*(Independent of frequency range)*

→ $R_i = R_{i,ref}\, f(T,SOC)$

*(Independent of T,SOC)*

# Features (Independent variables)

Raw

- Time, t
- Current, I
- Voltage, V
- Temperature, T

Derived

- Capacity, Q
- Resistance(T,SOC,I, pulse time or freq.)
- Max(T)
- $Q_{CC}$ or $t_{CC}$
- $Q_{CV}$ or $t_{CV}$
- $Q_{V0\text{-}to\text{-}Vf}$ or $t_{V0\text{-}to\text{-}Vf}$
- dQdV
  - Pos
  - Neg
  - Li inventory

What others are we missing?

- Acoustic
- Pressure/force
- Use history
  - Avg. temperature
  - Calendar age
  - Cycling throughput
  - Vehicle mileage

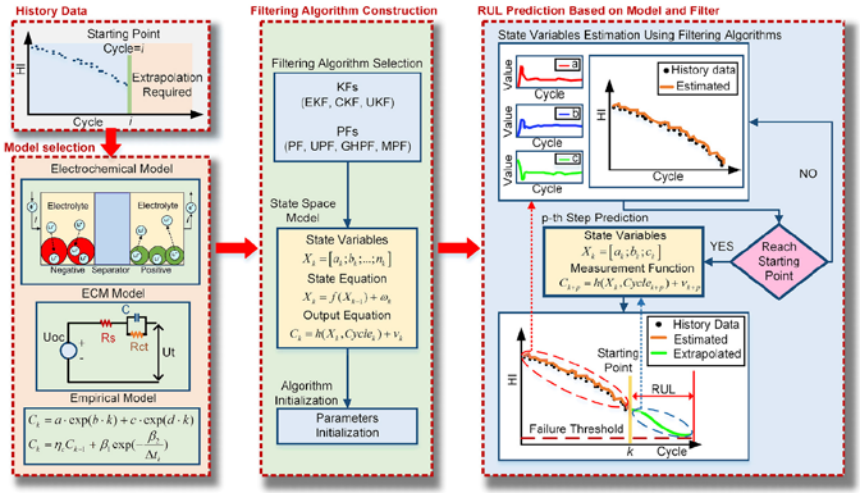
# Open-Source Aging Data

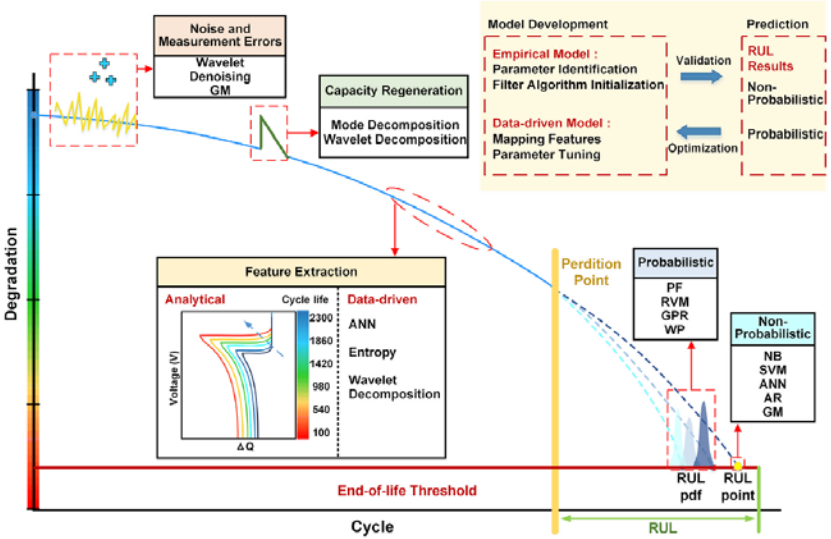
- NASA Ames, Prognostics Center of Excellence
  - Saha, B., and Goebel, K. (2007). Battery Data Set, NASA Ames Progn. Data Repos. NASA Ames Res. Cent. Moffett Field CA. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.
- U. Maryland, Ctr. For Advanced Life Cycle Engineering (CALCE)
  - He, W., Williard, N., Osterman, M., and Pecht, M. (2011). Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. J. Power Sources 196, 10314–10321.
  - Xing, Y., Ma, E.W.M., Tsui, K.L., and Pecht, M. (2013). An ensemble model for predicting the remaining useful performance of lithium-ion batteries. Microelectron. Reliab. 53, 811–820.
- Severson et al., MIT & Stanford
  - Severson, K.A., Attia, P.M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M.H., Aykol, M., Herring, P.K., Fraggedakis, D., et al. (2019). Data-driven prediction of battery cycle life before capacity degradation. Nat. Energy 4, 383–391.

# Extra

# Remaining Useful Life Prediction Review Article

Hu et al., Battery Lifetime Prognostics, Joule (2019), https://doi.org/10.1016/j.joule.2019.11.018




Remaining challenges in RUL prediction:

- Early life prediction
- Dynamic load conditions
  *Opportunity: Physics models*
- Computational complexity
  Statistical methods → AI methods (better nonlinear fitting)
  Empirical → Physics models (better accuracy)
  *Opportunity: Embedded control + Cloud computing; NN model reduction*
- Lack of data
  Researchers
  1st life → 2nd life
  *Opportunity: Blockchain + OpenData*

# PART 4 – Deep Learning / Neural Networks

Neural Network (NN) Theory

NN Inverse Models for Optimization (mesoscale, EIS)

NN Microstructure Segmentation

Generative Models

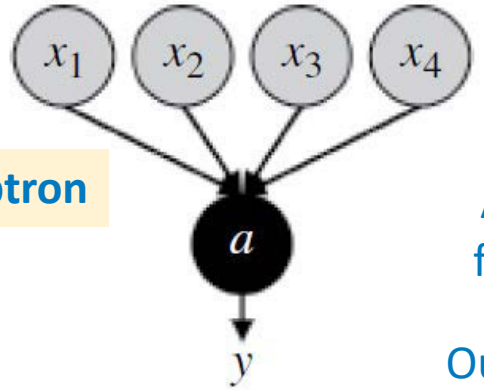
# Neural Network – Theory

# Deep Learning

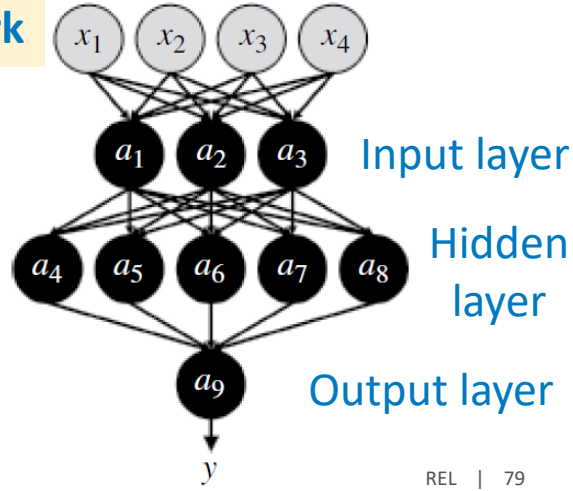
Figures: T. Mueller, Reviews in Comp. Chem. (2016)

- Mimic the biological brain
- Artificial neurons (a.k.a. nodes) linked like neurons
- "Deep" implies >1 hidden layer

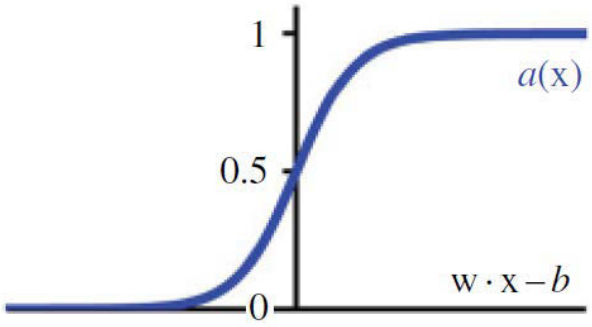
**Perceptron**




**Multilayer neural network**




**Sigmoid activation function, $a(x)$**




- Continuously differentiable (gradient descent requires derivative
- Works for classification (discrete), regression (continuous), and mixed problems

# NN Training via Backpropagation

Figure: T. Mueller, Reviews in Comp. Chem. (2016)
Equations: Andrew Ng/Coursera

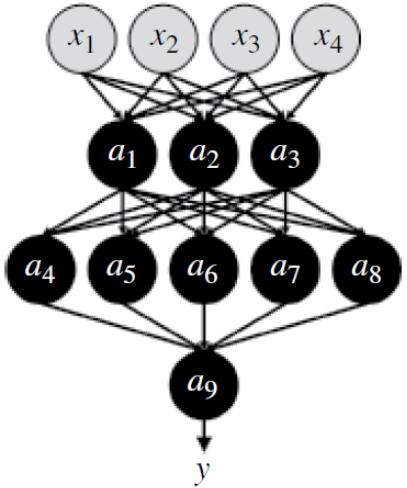
- Cost function

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(h_\Theta(x^{(i)})\right)_k + (1 - y_k^{(i)}) log\left(1 - \left(h_\Theta(x^{(i)})\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(\Theta_{ji}^{(l)}\right)^2$$

outputs; observations; layers; neurons in layer $l$

Gradient descent backpropagation algorithm

- For $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$,
- Initialize parameter matrix to random values, $-\epsilon \leq \Theta_{ij}^{(l)} \leq \epsilon$
- Step through data, for $i$ =1 to $m$
  - Set $a^{(i)} = x^{(i)}$
  - Forward propagate: Compute $a^{(i)}$ through all $L$ layers
  - Using $y^{(i)}$, compute $\delta^{(L)} = a^{(L)} - y^{(i)}$
  - Backpropagate: Compute $\delta^{(l)}$ for all other layers
  - $\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$
- $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) := \frac{1}{m}\Delta_{ij}^{(l)} + \lambda \Theta_{ji}^{(l)}$ if $j \neq 0$
- $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) := \frac{1}{m}\Delta_{ij}^{(l)}$ if $j = 0$
- * Note – gradients can also be calculated numerically. Analytical preferred, but numerical is useful to cross-compare to validate analytical result



# NN – Pros/Cons, Variants, Mat. Sci Applications

## Pros/Cons

- Pros: General applicability to linear/nonlinear, discrete/continuous, image/time-series, …
- Cons: Black-box model. Lacks interpretability. Data/time-intensive training

## Variants

- Architectures:
  - Dynamic systems: Recurrent NN include loops
  - Spatial systems: Convolutional NN
- Activation functions:
  - Sigmoid, tanh, rectified linear, softmax, swish
- Optimization methods

## Common materials applications

- Predict interatomic potentials
- Materials processing optimization
- Complex structure/property relationships

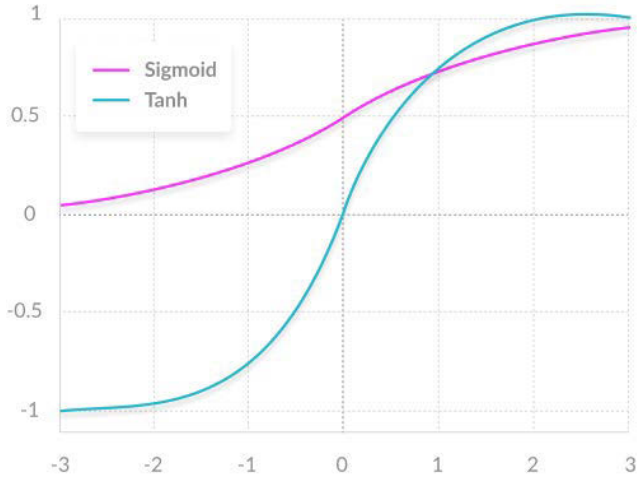



Figure: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

# NN Inverse Model

# Virtual Electrode Design via NN Inverse Model

*(Y. Takagishi, Batteries, 2019)*


batteries

MDPI

Article

Machine Learning Approaches for Designing Mesoscale Structure of Li-Ion Battery Electrodes

Yoichi Takagishi *, Takumi Yamanaka and Tatsuya Yamaue

Kobelco Research Institute Inc., Kobe 6512271, Japan

* Correspondence: takagishi.yoichi@kki.kobelco.com; Tel.: +81-78-992-5976

Received: 29 March 2019; Accepted: 23 July 2019; Published: 1 August 2019

check for updates

**Process parameters (predictor)**

Active material volume ratio
Particle radius
Conductivity of matrix
Binder/additives volume ratio
Pressure in compaction process

**3-D simulation physico-chemical simulations**

Generate electrode structures

Structure #1 Structure #2 Structure #3

Structure #4 Structure #5 ..... Structure #2100

Simulate each resistance factor

**Reaction resistance**

$$i = i_0\left[\exp\left(\frac{\alpha F}{RT}\eta\right) - \exp\left(-\frac{\alpha F}{RT}\eta\right)\right] \approx i_0\frac{2\alpha F}{RT}\eta$$

$$R_{reac} = \eta/(I \cdot L)$$

**Electrolyte resistance**

$$\nabla \cdot \left[-\sigma_l \nabla \phi_l\right] = 0$$

$$R_l = \left[\sigma_{l,eff}(1-\theta_b)^{1.5}\right]^{-1}$$

**Diffusion resistance**

$$\Delta c_s = i_s r^2/2D_s$$

$$R_{diff} = \left[\phi_{ocv}(c_0) - \phi_{ocv}(c_0 + \Delta c_s)\right]/(I \cdot L)$$

**Total resistance**

$$R_{tot} = R_l + R_{reac} + R_{diff}$$

**Regression**

Artificial Neural Network

Databse

**Electrode performance (Objective function)**

Total specific resistance

**Inverse problem**

Bayesian optimization

Figure: Takagishi et al., Batteries 2019, 5, 54

**Figure 1.** Proposed prediction and optimization scheme for a Li-ion battery porous electrode.

# NN Electrode Design – Data Generation

*(Y. Takagishi, Batteries, 2019)*

- Objective: Min. Total resistance = Reaction + Electrolyte + Diffusion
- 2100 simulations of virtual generated microstructures
- Selected features
  - Active volume fraction
  - Particle radius
  - Pressure
  - Binder/additive vol. frac.

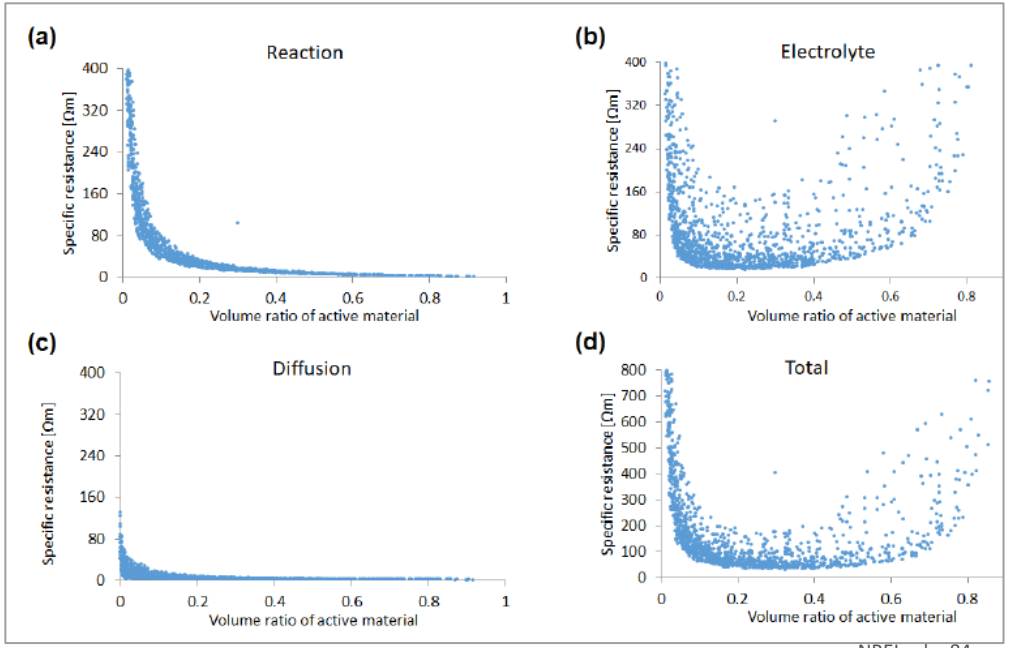



Figure: Takagishi et al., Batteries 2019, 5, 54

# Electrode Design – NN Selection

*(Y. Takagishi, Batteries, 2019)*

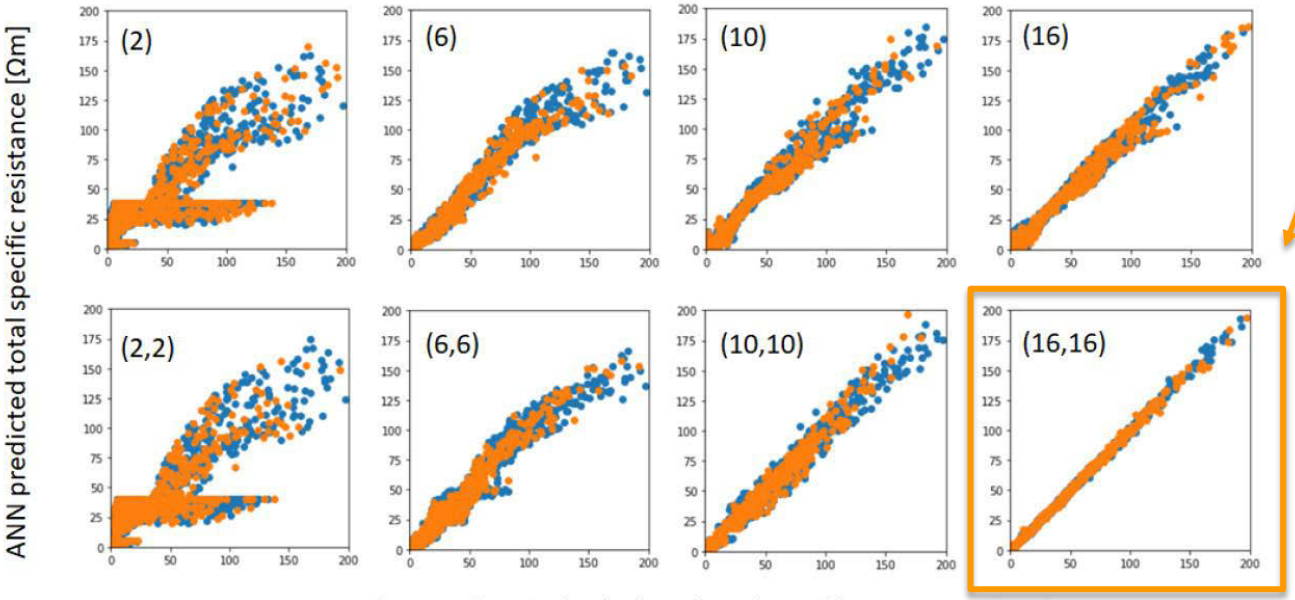
Table & Figure:
Takagishi et al.,
Batteries 2019, 5, 54

Table 4. The correlation coefficients, $R^2$, for the validation data.

| Number of Hidden Layers | Number of Neurons | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | 4 | 6 | 8 | 10 | 12 | 16 | 20 |
| 1 | 0.326 | 0.788 | 0.834 | 0.810 | 0.790 | 0.978 | 0.937 | 0.979 |
| 2 | 0.330 | 0.795 | 0.822 | 0.863 | 0.827 | 0.845 | 0.990 | 0.990 |

Selected NN based on $R^2$ saturating at 0.990

- 2 hidden layers &
- 16 neurons



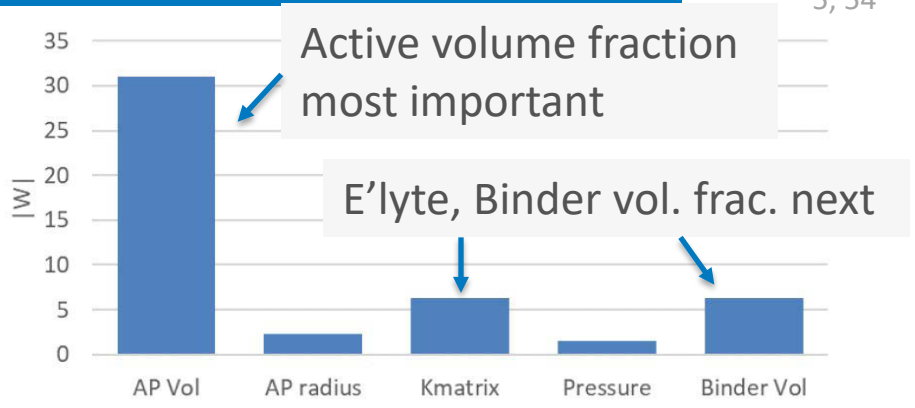

- Training data (70%)
- Validation data (30%)

# Electrode Design – Ranking Feature Importance

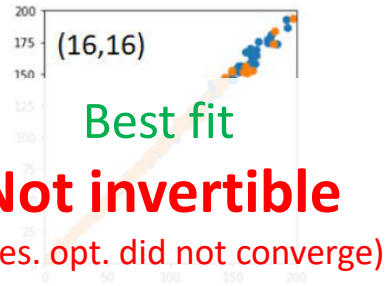
*(Y. Takagishi, Batteries, 2019)*

Figures: Takagishi et al., Batteries 2019, 5, 54

- 1st-layer-neuron weight-coefficients are summed vs each design parameter




## Optimal design
(Inverse problem)



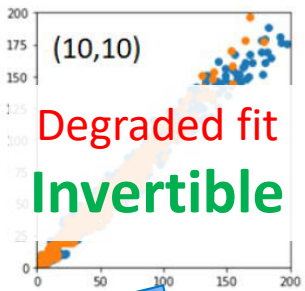




Table 5. Optimized process parameters for [illegible]otal specific resistance.

| Active Material | | Binder/Additives | Electrolyte | Compaction |
|---|---|---|---|---|
| Volume Fraction (%) | Radius (μm) | Volume Ratio (%) | Conductivity (S/m) | Pressure (MPa) |
| 50.4 | 6.00 | 0.0820 | 1.00 | 590 |

## Future:

- +Experiments, validation
- Energy density co-optimization
- Convex Neural Network (invertible)

# EIS Fitting Using NN to Provide Initial Parameter Guesses

- EIS data notoriously hard to fit
  - Dependent on human initial guesses
- Trained NN to provide initial guesses

Journal of The Electrochemical Society, 166 (8) A1611-A1622 (2019)

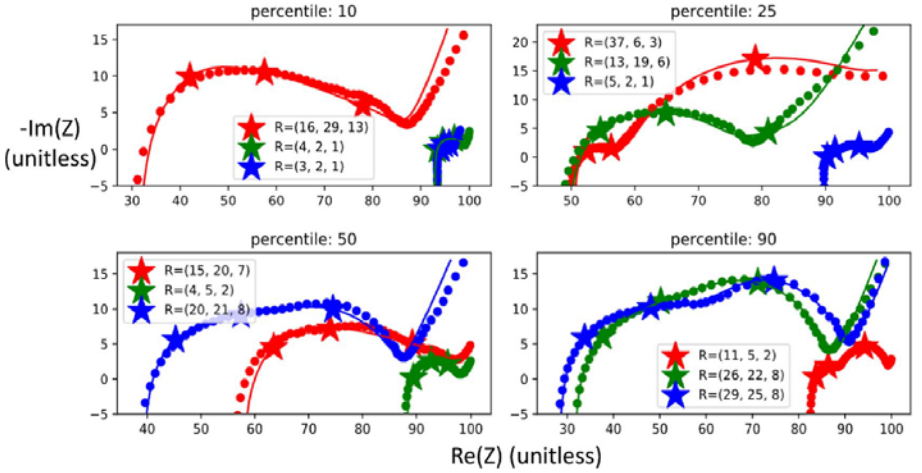
**Analysis of Thousands of Electrochemical Impedance Spectra of Lithium-Ion Cells through a Machine Learning Inverse Model**

Sam Buteau[1,*] and J. R. Dahn[1,2,**,z]

[1]Department of Physics and Atmospheric Science, Dalhousie University, Halifax, NS B3H 4R2, Canada
[2]Department of Chemistry, Dalhousie University, Halifax, NS B3H 4R2, Canada

Electrochemical impedance spectra of lithium-ion cells can be collected periodically at various cycle numbers and various state of charges, producing vast amounts of data. Fitting each spectrum to an equivalent circuit can lead to physical insights about the evolution of the lithium-ion cell, yet the fitting problem requires good human initial guesses for the circuit parameters to reliably converge, making the fitting process labor intensive and difficult to scale. This article presents a paradigm to automate the fitting of measured data to physical models, replacing the good human first guesses with an inverse model parametrized with an artificial neural network. This method is simple to implement, uses principles applicable to a wide variety of fitting problems, and leads to reliable and accurate initial guesses of the circuit parameters for a given spectrum. The software implementation will be freely available once a good user interface is developed, and the performance of the system is evaluated on a dataset of about 100000 impedance spectra from lithium-ion cells, achieving a failure of fitting approximately 1% of the dataset, corresponding to the percentage of poor quality data in the dataset.

| Component Name | Circuit Representation | Formula |
|---|---|---|
| Resistor | $R$ | $Z_{\text{ohm}}(\omega) = R$ |
| CPE | $(Q/\varphi)$ | $Z_{\text{CPE}}(\omega) = \frac{1}{Q(i\omega)^{\varphi}}$ |
| ZARC | $(R/\omega_c/\varphi)$ | $Z_{\text{ZARC}}(\omega) = \frac{R}{1+(i\frac{\omega}{\omega_c})^{\varphi}}$ |




Figures: S. Buteau, J. Echem Soc, 2019.

# Microstructure Segmentation – Fiji WEKA Software

# WEKA (Waikato Environment for Knowledge Analysis)

The **Trainable Weka Segmentation**[1] is a Fiji plugin that combines a collection of machine learning algorithms with a set of selected image features to produce pixel-based segmentations.

The advantages of Weka include[2]:

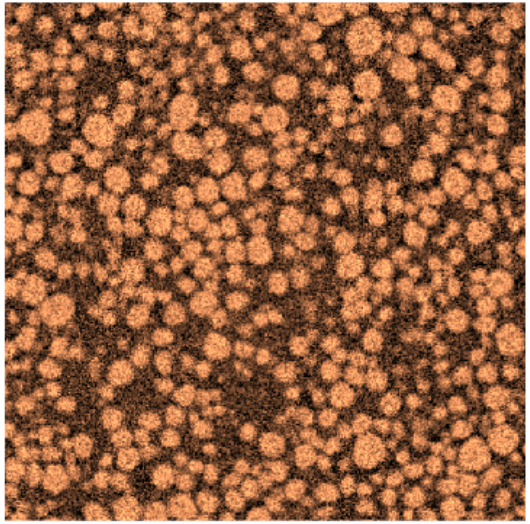
- freely availability under the GNU General Public License
- portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform
- a comprehensive collection of data preprocessing and modeling techniques
- ease of use due to its graphical user interfaces

[1] Arganda-Carreras, I.; Kaynig, V. & Rueden, C. et al. (2017), "Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification.", *Bioinformatics* (Oxford Univ Press) **33** (15), PMID 28369169, doi:10.1093/bioinformatics/btx180 (on Google Scholar).

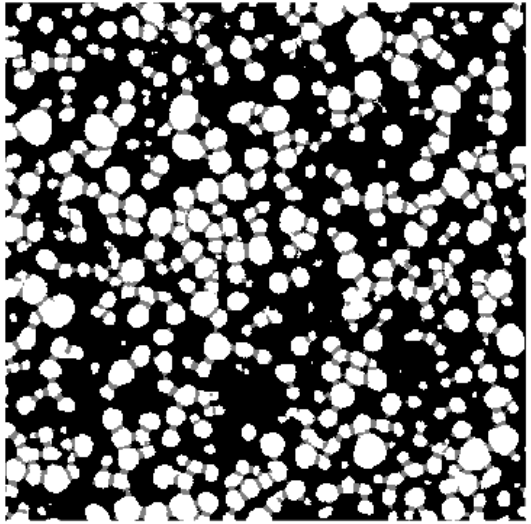
[2] https://imagej.net/Trainable_Weka_Segmentation

# Example 1: Identifying NMC particles (2D image)

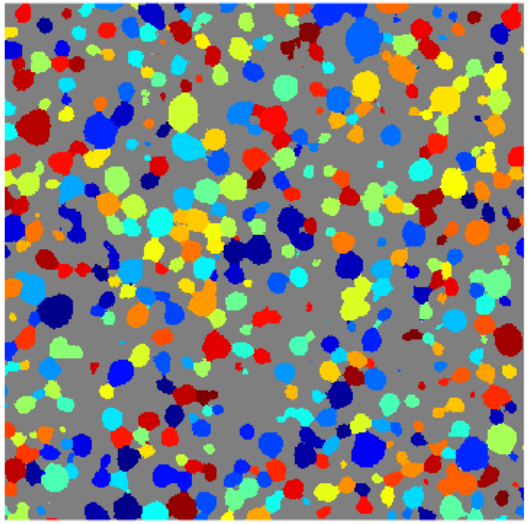
Grey level image



Pores, particles, particle-to-particle segmentation
WEKA machine-learning



Particles labelling
MATLAB connectivity analysis



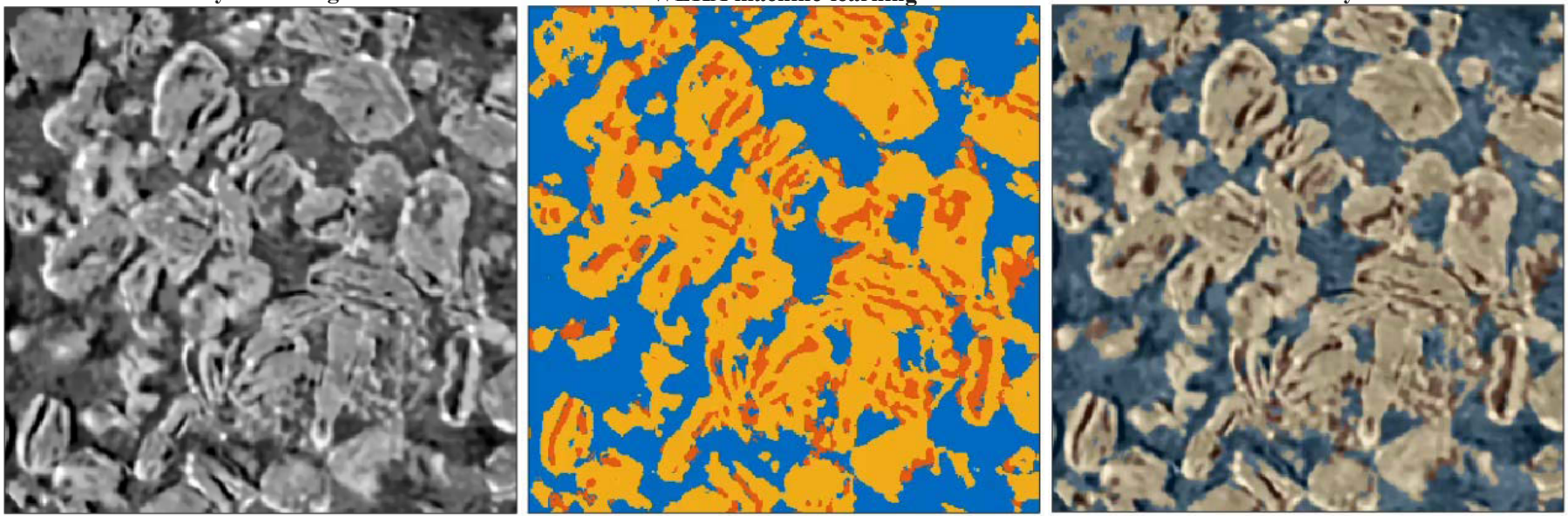
- Image quality is good enough not to require advanced method such as machine learning to identify pore and solid domain. However, threshold-based segmentation provides a nearly fully connected solid domain. This example shows it is possible **to identify individual particles** through machine learning
- Classifier trained with 3 phases: pores, particles, **particles connection**.
- From this segmentation, a simple connectivity analysis can easily distinguish each particles individually
- **Enable particle morphology and size analysis**

# Example 2: Identifying open-porosity cracks/voids in graphite (3D volume)

Grey-level image

Pores, cracks, particle segmentation
WEKA machine-learning

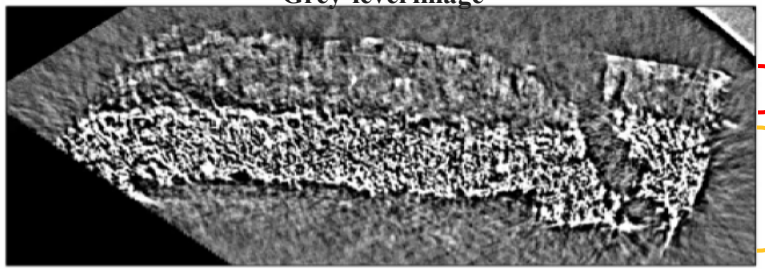
Overlay



- Distinction between the background (pore) and cracks is ambiguous. Threshold-based method will be highly subjective. Even though some cracks look like closed-porosity, they are actually connected to the pore in 3D, thus connectivity analysis cannot distinguish pores from cracks.
- Classifier trained on the 3D volume with 3 phases: pores, **cracks**, particles.
- **Enable quantification of crack density, and specific surface area due to cracks**

# Example 4: Segmenting multi-phase, highly convoluted graphite/Li image (3D volume).

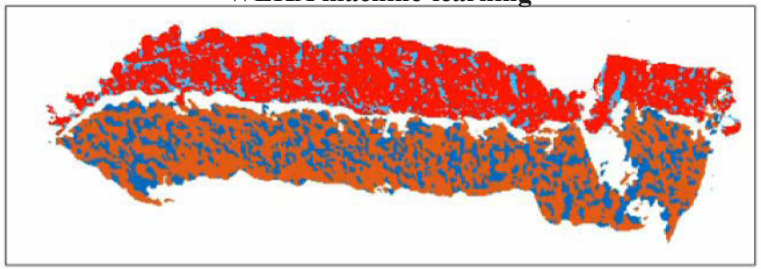
Grey-level image



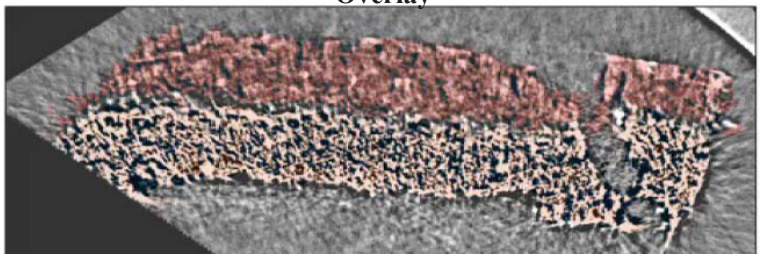
Plated Li metal

Graphite electrode

4 phases + background segmentation
WEKA machine-learning



Overlay



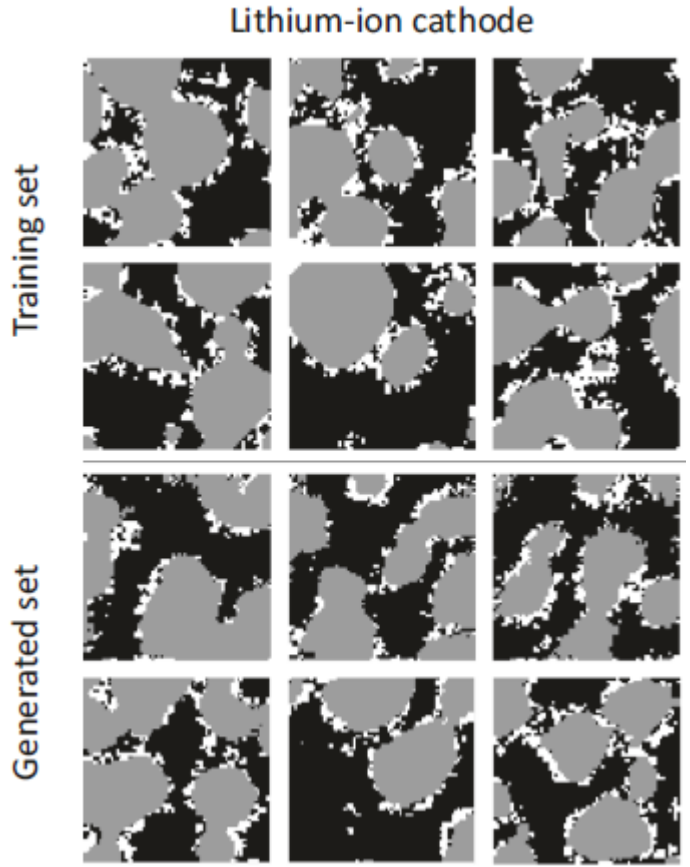
- Difficulty comes from convoluted image and similar grey level value between several phases (pores of both layers, background and metal deposition solid phase).
- Threshold-based method cannot segment this image.
- **Classifier trained on 5 phases: solid and pore of both layers, plus the background**
- **Enable quantification of metal deposition**

# Other

## Generative Adversarial Network

## Safety Map

# Generative Adversarial Network – Microstructure Generation




npj | Computational Materials
www.nature.com/npjcompumats

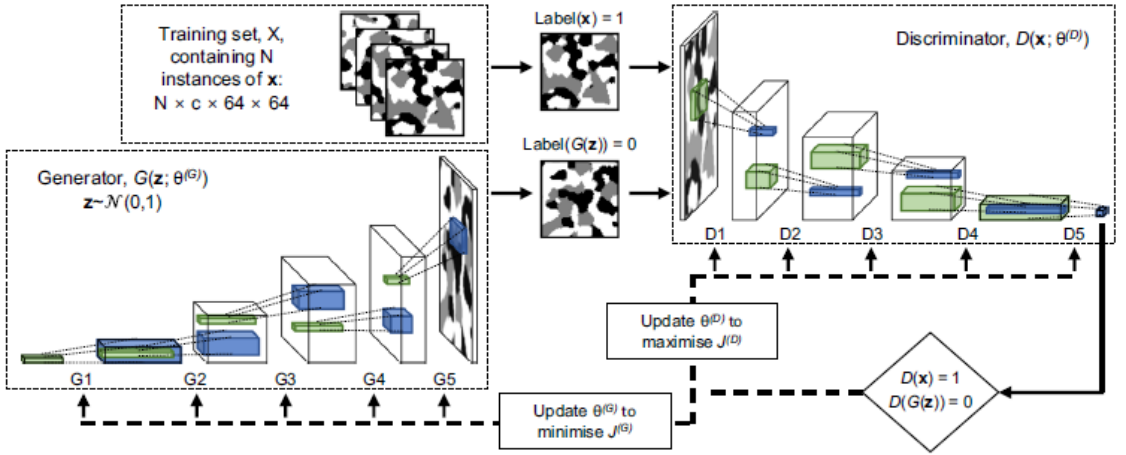
ARTICLE OPEN

Check for updates

## Pores for thought: generative adversarial networks for stochastic reconstruction of 3D multi-phase electrode microstructures with periodic boundaries

Andrea Gayon-Lombardo[1], Lukas Mosser[2], Nigel P. Brandon[1] and Samuel J. Cooper[3]

The generation of multiphase porous electrode microstructures is a critical step in the optimisation of electrochemical energy storage devices. This work implements a deep convolutional generative adversarial network (DC-GAN) for generating realistic n-phase microstructural data. The same network architecture is successfully applied to two very different three-phase microstructures: A lithium-ion battery cathode and a solid oxide fuel cell anode. A comparison between the real and synthetic data is performed in terms of the morphological properties (volume fraction, specific surface area, triple-phase boundary) and transport properties (relative diffusivity), as well as the two-point correlation function. The results show excellent agreement between datasets and they are also visually indistinguishable. By modifying the input to the generator, we show that it is possible to generate microstructure with periodic boundaries in all three directions. This has the potential to significantly reduce the simulated volume required to be considered "representative" and therefore massively reduce the computational cost of the electrochemical simulations necessary to predict the performance of a particular microstructure during optimisation.



# Cell Crush Safety Map via Physics Simulation & ML

*(Li, Joule, 2019)*

Joule

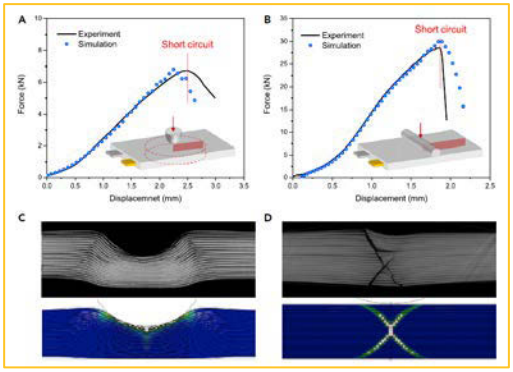
Joule 3, 2703–2715, November 20, 2019 © 2019 Elsevier Inc. 2703

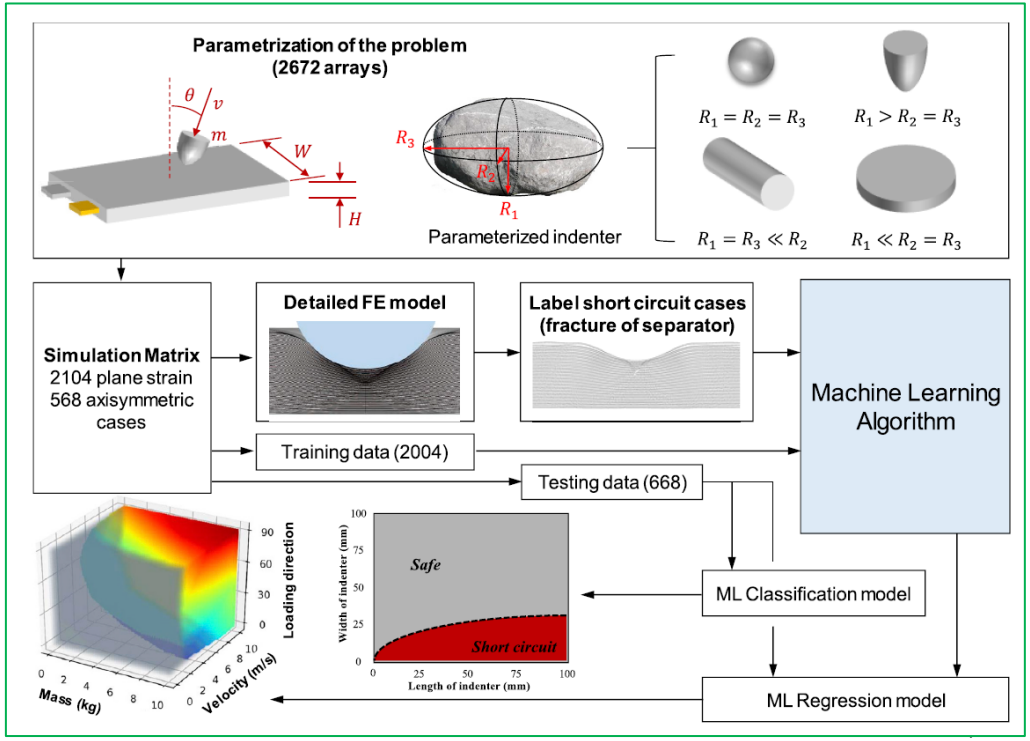
Article

## Data-Driven Safety Envelope of Lithium-Ion Batteries for Electric Vehicles

Wei Li,[1,2,3] Juner Zhu,[1,3,4,*] Yong Xia,[2,*] Maysam B. Gorji,[1] and Tomasz Wierzbicki[1]

Detailed finite element model of mechanical crush




- Predict separator failure from deformed geometry
  - NN w/ 1 hidden layer & 50 nodes
- Generate safety map
  - SVM with radial basis function (RBF) kernal

# Conclusions & Outlook

- Is the algorithm, right? (linearity, error assumptions, architecture)
  - How can I incorporate the physics of the problem?
- Do I have enough data? Do I have the right data? Will the model extrapolate?
- Center and scale! (eat your vegetables)

Applications:
- Materials discovery
- Microscopy, spectroscopy
- Performance, lifetime, safety
- Manuf. process opt.
- Real-time control
- Fleet management

"Crowd-sourced" data
- Materials databases
- DFT
- Complex, hierarchical data
  - Echem
  - Microscopy
  - Spectroscopy
- 3D continuum simulations
- Real-world data

Paths forward
- Open-source data
  - Lab & real world
  - Diversity of time & length scales
- Challenge problems
- Validity of ML algorithms across chemistries, designs

# Acknowledgements


- DOE Office of Vehicle Technologies
  - Samuel Gillard, Energy Storage Technology Manager
  - Simon Thompson, Energy Storage Technology Manager
  - Steven Boyd, Vehicle Electrification Program Manager
  - Michael Berube, Director
  - David Howell, Deputy Director

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Vehicle Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.


Thank you!
https://www.nrel.gov/transportation/energy-storage-publications.html
Email: kandler.smith@nrel.gov