The generalisation challenge: assessment of the efficacy of acoustic signals for state estimation of lithium-ion batteries via machine learning

Elias Galiounas[a], Rhodri E. Owen[a,b,c], James B. Robinson[a,b,c], Rhodri Jervis[a,c,*]

[a] *Electrochemical Innovation Lab, Department of Chemical Engineering, University College London, London, UK, WC1E 7JE,*
[b] *Advanced Propulsion Lab, Marshgate, University College London, London, UK, E20 2AE,*
[c] *The Faraday Institution, Quad One, Harwell Science and Innovation Campus, Didcot, UK, OX11 0RA,*

Abstract

Acoustic measurements of batteries are known to be correlated to their state-of-charge, creating opportunities for state estimation that do not rely on electrical signals. State estimators are typically parametric models fitted from data, often from the broad toolbox of machine learning. Such models can be easily designed to have millions of tuneable parameters, which endows them with tremendous but often misinterpreted fitting ability. The real performance metric, commonly omitted in the battery literature, is a model's generalisation performance with respect to a population, which requires successful predictions to be made on data from one or more 'held out' cells. This study demonstrates that regression models based on neural networks can perform highly accurate state estimation on multiple cells; however, this is shown to be conditional on all cells being represented in the training dataset. Generalisation to the wider population is shown to be more challenging than other studies claim; a conclusion which follows from tests on multiple feature configurations and multiple model variants. It is hypothesized that success on multi-cell data in the absence of wider generalisation is due to the ability of models to learn cell-specific patterns implicitly, which is a type of 'overfitting'. This hypothesis is tested in two ways. First, classifiers performing a matching operation between acoustic waveforms and their respective cells are used to show that cell-specific characteristics are present in the waveforms. Next, unsupervised learning methods are used to perform a projection of all acoustic signals to two-dimensional latent space. In the latent space it is found that datapoints cluster according to the cell identity, indicating that the distinctiveness of cells dominates over any state-related commonalities in the acoustic dataset. The study highlights the need for caution in how the generalisation of machine learning models (of any kind) is evaluated in battery research.

Keywords:
Ultrasonic battery monitoring, Battery diagnostics, State-of-charge, Machine learning, Battery populations, Acoustic testing

*Corresponding author: rhodri.jervis@ucl.ac.uk

1. Introduction

The properties of acoustic waves, such as their speed and attenuation, are influenced by the mechanical properties and geometry of the medium they traverse. In the case of lithium-ion batteries, this manifests as a correlation to the state-of-charge (SoC), since the mechanical properties and geometry of electrodes vary and evolve with the state. Demonstrations of this behaviour were made definitively in the past decade [1, 2] and were followed by attempts to leverage the correlation in constructing state estimators reliant on acoustic signals. The aim was never to replace conventional methods of SoC estimation, which can be highly successful (such as electrochemical-model-based approaches [3, 4]), but rather to complement and diversify them, targeting improvements in safety and reliability. The potential for such improvements is based on the independence of the acoustic signal, which is measured by a dedicated sensor and is also uniquely informed by the internal structure and chemomechanics of a cell, rather than its aggregate electrochemical characteristics. Therefore, acoustic state estimation can be performed separately from other methods, and can be used to verify them, detect faults, or contribute towards the quantification of uncertainty.

Much of the existing literature is focused on reducing waveforms to specific features, investigating their individual dependencies on the SoC, and possibly on additional parameters such as the temperature and C-rate. Gold et al. [5] performed tests at relatively low ultrasonic frequencies, near 200 kHz, where they observed the separation of travelling waveforms into a fast and a slow wave according to Biot theory [6, 7]. They then argued in favour of using features of the slow wave and demonstrated state estimation by linear regression. Wei et al. [8] also searched for a feature that would be linear in the SoC, and their recommendation was termed the 'initial rise time' — the time between the front and the centroid of the transmitted signal.

Zhang et al. [9] developed a methodology to extract six features from a certain interval of the time domain, where the interval itself was selected based on correlation metrics to the SoC. Li et al. [10] extracted eleven features using both the time and the frequency domain, and then downselected seven of those, also based on their correlation and sensitivity to the SoC. Galiounas et al. [11] had previously demonstrated that the frequency domain contains useful structures and that time-domain features could be filtered down. In a study on cylindrical cells, Montoya-Bedoya et al. [12] extracted parameters from the power spectrum, highlighting a particular parameter called the 'mid-band fit'. This was computed by performing a line-fitting operation on the normalised power spectrum which, according to the authors, is a method used in medical applications.

The above studies advocate for their respective feature extraction technique based on criteria such as stability, monotonicity or linearity of the correlations to the SoC. Such characteristics have been shown to be present in any one cell; nevertheless, they have not been shown to be consistent in a population so as to allow a generalised SoC estimator to be fitted for a specific battery product (i.e. for a catalogue-listed cell). In cases where SoC models were in fact fitted, the underlying dataset typically consisted of signals obtained at a single location of a single cell [11, 9, 10, 13, 14, 5, 8]. This monolithic nature of the datasets has resulted in a variety of modelling techniques being successful in estimating the SoC of a single cell, and various types of artificial neural networks have performed particularly well [11, 9, 10, 13]. Nevertheless, in the absence of a benchmark test, different methods cannot be compared and the accuracies reported by different studies are not worth noting. Furthermore, single-location testing is prone to producing datasets that correlate acoustic waveforms to the local SoC of the tested region, instead of the global

SoC of the whole cell. Spatial variations of the SoC, especially in large-format cells, have been demonstrated by several diffraction studies [15, 16, 17, 18].

Huang et al. [19] deviated from the single-cell single-location paradigm in two ways. Firstly, they trained an SoC estimator using waveforms from 150 locations of one cell and 42 SoCs (that amounts to 6300 waveforms). Secondly, as an assessment of generalisation they used their trained estimator to infer the SoC of two other cells, using signals from 6 different locations on each. A point of novelty is that they did not compare their predictions to the aggregate SoC of those cells, but to the local SoC of those specific points, measured destructively by inductively coupled plasma-induced optical emission spectroscopy (ICP-OES). Although this generalisation test cannot be considered extensive, due to the small number of spectroscopic test points, it would not be practical to perform ICP-OES much more widely. A complementary assessment could have evaluated the trained model on more locations of the test cells, comparing predictions to the cell-level SoC. The authors had 141,750 such datapoints from each cell in their disposal (obtained from 3,375 locations and 42 SoCs), and additional model validation using this dataset would be useful in the future. Davies et al. [20] also deviated from the single-cell paradigm by training a Support Vector Machine with acoustic data from 2 cells and predicting the SoC of a third cell, although the size of the dataset and the number of training and test samples were not specified in their study.

None of the studies discussed above have published their datasets; therefore, it is not possible to conduct comparative evaluations or to validate claims. In this work we share an experimental dataset containing acoustic signals from 7 cells cycled using a multi-C-rate protocol. We demonstrate visually that acoustic signals from different cells are heterogeneous when simple features are considered, and proceed to explore whether different feature configurations can result in greater levels of homogeneity across cells and lead to state estimators that can successfully generalise. State estimators are constructed in the form of feedforward neural networks (FNNs) and convolutional neural networks (CNNs), whose input is a certain feature configuration and whose output is the cell voltage. This is a regression task. For each feature configuration, estimators with a range of fitting capabilities are trained and tested.

Generalisation is shown to be challenging, and the study proceeds to investigate why. It is shown that very simple classification models can correctly identify which cell produced an acoustic signal, indicating that certain acoustic characteristics are cell-specific. Lastly, a breadth of unsupervised learning techniques is employed to reduce the dimensionality of acoustic signals to two latent dimensions. This aims to investigate the possible emergence of clustering-by-cell in the latent space, which would indicate the prevalence of cell distinctiveness over state-related commonality in the acoustic dataset.

The data and models generated in this work are shared open access [21], as well as the code used in processing, visualisation and animation of the data. This code is in the form of a purpose-built python package titled SonicBatt [22]. The reader can follow the instructions in the SonicBatt repository to reproduce most plots of this study.

2. Experimental methods

2.1. Data generation

A total of seven commercial cells with a nominal capacity of 210 mAh and $LiCoO_2$/Gr chemistry (Model 651628, AA Portable Power) were used. A complete specification sheet is provided in the Supplementary Information (SI). The same cell was used in some of

the aforementioned acoustic studies [20, 11], including Davies et al. [20] who claim a generalised state estimator. Cycling was performed with a computer-controlled potentiostat (Interface 1010E, Gamry instruments, US). The surface temperature was monitored with an N-type thermocouple, attached to the surface of cells using polyamide adhesive tape, and using a TC-08 thermocouple interface (Pico Technology, UK). Tests were conducted on a bench top and the ambient temperature varied according to laboratory conditions. All cycling was performed between the manufacturer's stated voltage limits of 2.75 and 4.2 V.

The cycling protocol included five repetitions of the following sequence: (1 × 0.2C), (3 × 0.5C), (5 × 1C). Therefore, a complete cycling protocol included 45 cycles. Fig. 1a demonstrates the cycling protocol for Cell 1 out of 7, and the equivalent data for Cells 2–7 can be found in SI Section 1. Acoustic data was acquired using an Olympus Epoch 650 ultrasonic flaw detector and a single 6.35 mm diameter, 5 MHz transducer operating in pulse-echo mode (M110-RM, Evident Scientific). The transducer was secured by a 200 g weight. The ultrasonic settings used are specific to the Olympus detector and are listed in Table 1. A small amount of silicon-dioxide-based couplant (H-2, Evident Scientific) was applied between the transducer and the cell. The shape of the pulsed waveform, obtained by conducting tests on aluminium, can be found in our previous work [23].

Acoustic waveforms were recorded every 60 seconds while cycling. This produced a dataset of 66,995 waveforms, with similar contributions from the different cells (Table 2). An example waveform is shown in Fig. 1b. It comprises a total of 4000 data points which correspond to a duration of 10 µs. The 9 peaks circled on the waveform are peaks that could be identified consistently for all cells under all cycling conditions thanks to their adequate prominence. They will later be used to form certain configurations of acoustic features. The final peak is widely believed to originate at the posterior side of the cell [24] and can be termed the 'back wall echo' peak. As a convenient feature to visualise, the time-of-flight (ToF) of this 'back wall echo' peak is also plotted in Fig. 1a.

Table 1: Ultrasonic flaw detector settings.

Parameter	Value
Energy Gain	300 V
Gain	51 dB
Range	10 µm
Filter	0.5–4 MHz
Pulse Frequency	2.25 MHz

2.2. Dataset visualisation with example acoustic features

Cycling and acoustic information for the entire dataset are plotted in Fig. 2 versus the cell charge level (Q). Different colours are used for the different C-rates, and lighter shades represent earlier cycles than deeper shades. Voltage is plotted in the first row and is representative of the electrochemical characteristics of the cells. The remaining rows demonstrate acoustic features which characterise the acoustic response. Namely, the amplitude and the ToF of the second and the last acoustic peaks are shown, as well as the difference between those values from one peak to the other. The second rather than the first peak was chosen for visualisation because the first peak saturated in certain cases (it overshot the maximum value that could be recorded).

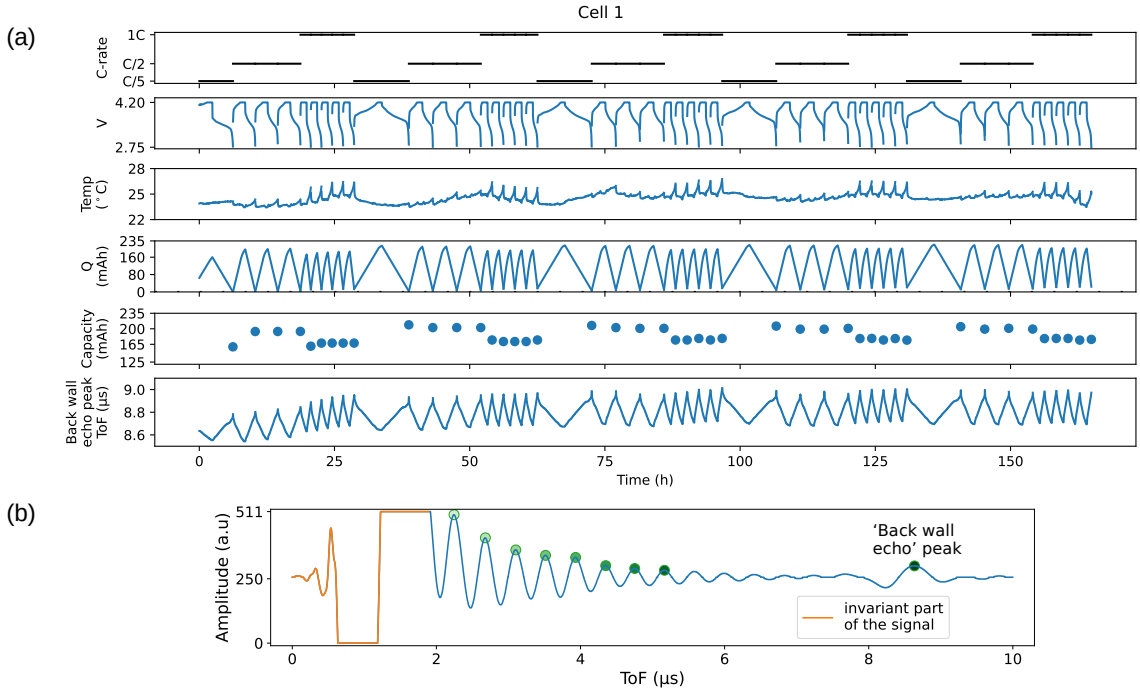


Figure 1: (a) Example cycling protocol for Cell 1 out of 7, including the sensed voltage and temperature signals, capacity measurements obtained by Coulomb counting, and the 'back wall echo' ToF of the acoustic signals. (b) Example acoustic waveform with identified peaks.

Table 2: Population of acoustic signals in the dataset (per cell and total).

Cell ID	Number of signals
1	9,917
2	10,105
3	9,034
4	8,828
5	10,438
6	9,158
7	9,515
Total	**66,995**

It is worth noting the hysteresis that is present in most acoustic features, which creates 'boxy' shapes similar to the voltage plots. Voltage hysteresis is a common phenomenon in lithium-ion batteries, and is more pronounced at high currents [25, 26]. The acoustic hysteresis observed here shows a similar dependence on the current as voltage hysteresis, especially when the acoustic amplitude is concerned. The ToF of the last acoustic peak (the 'back wall echo' peak) exhibits less hysteresis during cycling and is relatively linear, providing a compelling parameter for state estimation by linear regression or similar methods [24]. Nevertheless, it is not a consistent feature between different cells, and would therefore require cell-specific model parameterisation. In fact, none of the other acoustic features shown are adequately consistent between all cells. This motivates the use of richer feature configurations for the task of state estimation, together with more advanced models that are capable of capturing non-linear patterns. The next section will

proceed in this direction.

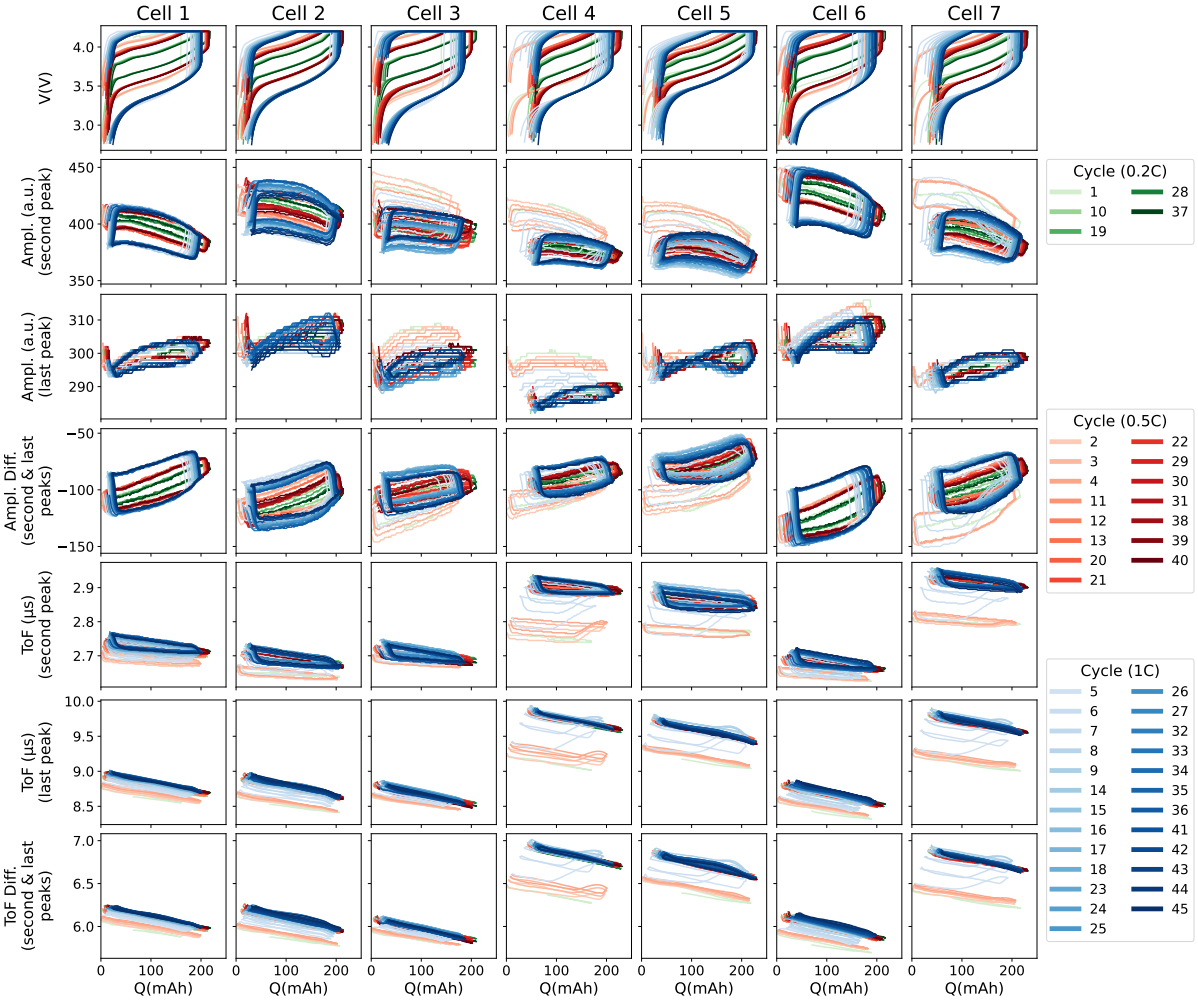


Figure 2: Summary plot of entire dataset. Voltage and acoustic data plotted against the amount of cell charge (Q). Three C-rates: 0.2C, 0.5C, 1C. Light shades indicate early cycles. Darker shades indicate later cycles. Similar plots for the individual C-rates can be found in SI Section 2.

A challenge revealed by Fig. 2 is the presence of some acoustic instability at the beginning of each test. Early cycles (light shades) appear in many cases to produce an acoustic response that rapidly evolves before stabilising. Whether this acoustic instability implies that a cell's electrochemical characteristics are unstable should be the subject of further work, although the investigations of Knehr et al. [27] are instructive and were conducted on the same cell. Some electrochemical instability can in fact be seen in the individual cycling profiles of the seven cells (SI Section 1), where capacity measurements obtained by Coulomb counting indicate that some lithium-consuming processes similar to formation are still taking place in the early stages of cycling. Consequently, the charge level (Q) shown as the x-axis in Fig. 2 is not perfectly accurate because Coulomb counting accounts for all charge passed, including irreversible charge lost to parasitic reactions. Additionally, Coulomb counting is prone to accumulation of rounding errors over time, and each experiment had a duration of approximately 1 week which can make rounding errors problematic. For these reasons, the state estimators trained in the next sections will use voltage as their target variable — the sole dependent variable in the regression

tasks.

3. Computational methods

3.1. Feature configurations and data splits

Acoustic waveforms were formulated into seven feature configurations labelled 'A' to 'G'. The feature configurations are described visually in Fig. 3 and verbally in Table 3, which also lists the number of features in each configuration. Config 'A' is the simplest, capturing the x-y position of the 'back wall echo' peak, and will be used as a benchmark. Configs 'B' and 'C' contain information from additional peaks. Config 'D' uses the entire time domain of each waveform excluding the first 1.9 μs segment, approximately, which is invariant in the dataset. Config 'E' is obtained by performing an FFT on config 'D', and keeping the magnitude spectrum of the first 300 frequency bins (0.12–37.0 MHz). This frequency range is slightly different compared to the previous chapter, because of the cropping of the first 1.9 μs that was applied to the time-domain. Config 'F' attempts to combine time- and frequency-domain information by concatenating configs 'D' and 'E' into one vector (not visualised in Fig. 3 because the two domains have very different scales). Config 'G' has the same aim, but instead converts waveforms into spectrograms which reveal frequency changes over time and can be visualised as 2D images.

Spectrograms were computed by performing a Short-Time Fourier Transform (STFT) over each waveform. This involves windowing the time domain, computing the FFT for the windowed signal, and then stepping the window across. The following design choices were made for the creation of spectrograms: A Hann window with a length of 501 datapoints was used, corresponding to a signal duration of 1.2525 μs (whole signals are 10 μs long). The step size was 5 data points (12.5 ns) and no padding was used; therefore, the window was stepped across 549 times. Among the calculated frequency bins only the first 20 were kept (0.8–16.0 MHz). These design choices produced spectrograms of manageable size (549 × 20 arrays) for the subsequent training tasks. Also, the resulting spectrograms were animated and visually inspected, and were found to vary visibly with cycling (bins of higher frequencies, above 16 MHz, did not). Animations of the spectrograms, and of the time and frequency domains of all signals and all cells, can be downloaded from the online data repository [21].

Table 3: Descriptions of the seven feature configurations and the number of features in each.

Feature config	Description	Number of features
A	'Back wall echo' peak ToF & Amplitude.	2
B	9 peaks ToF.	9
C	9 peaks ToF & Amplitude.	18
D	Time domain (except invariant part).	3242
E	Frequency domain obtained from feature config 'D'. — First 300 frequencies (0.12–37.0 MHz).	300
F	D & E together.	3542
G	Spectrograms obtained from feature config 'D'.	10980

For machine learning tasks, the 7-cell dataset was split into training, validation, and test sets. This splitting was carried out in two different ways, as outlined in Table 4, in

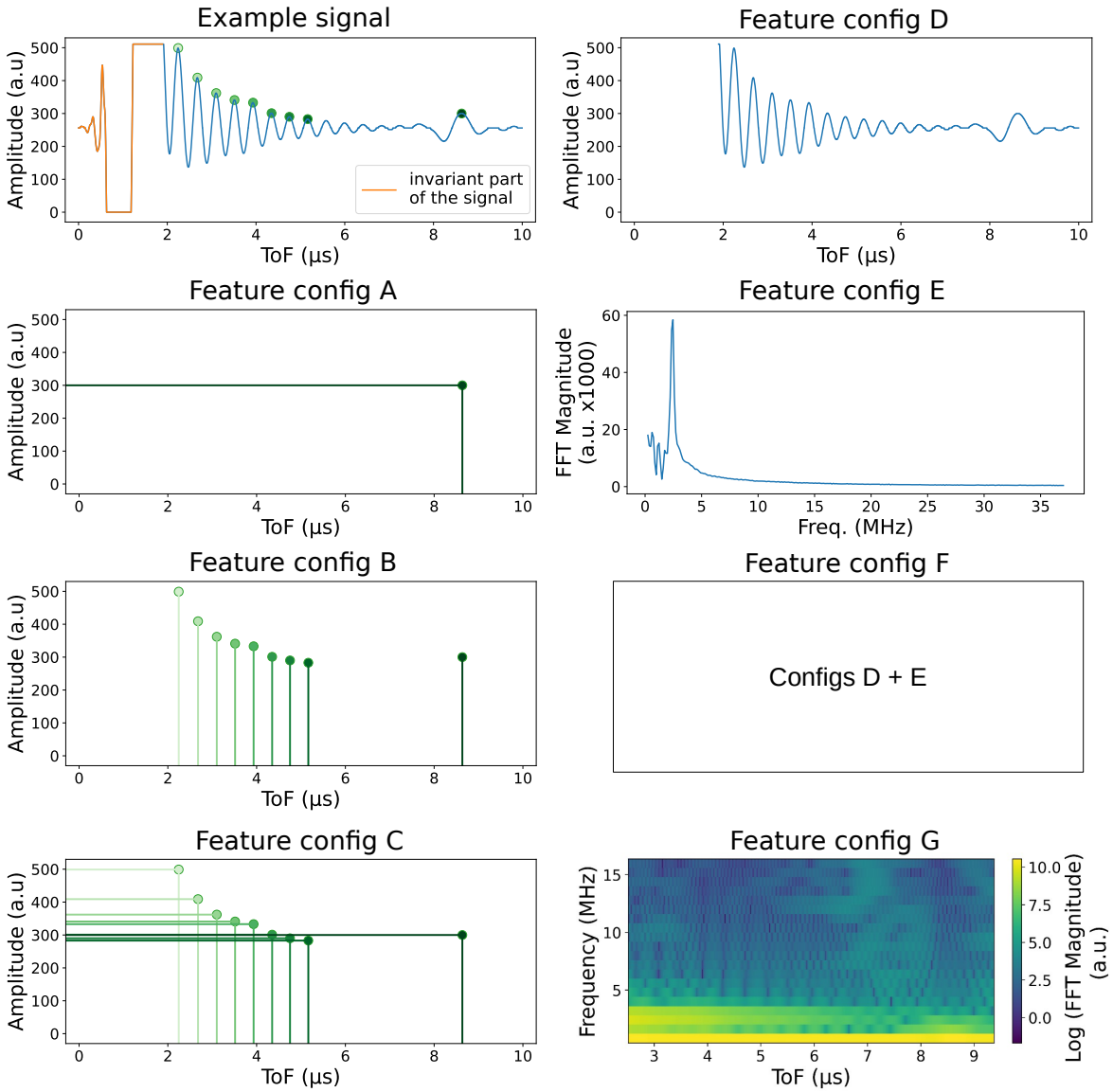


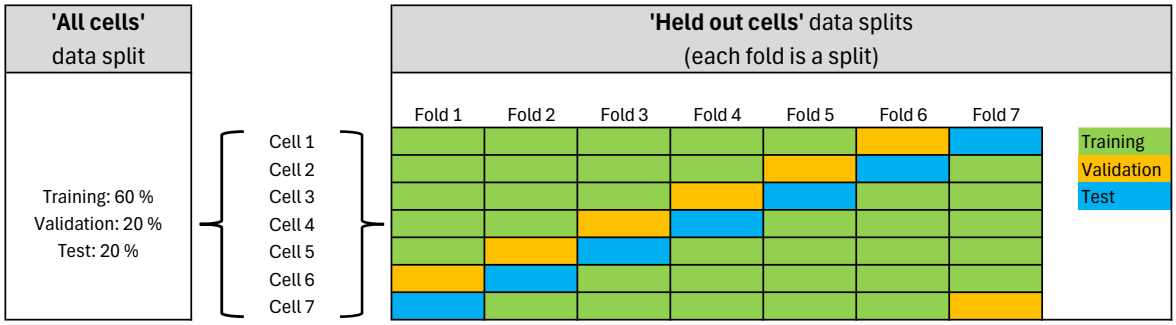
Figure 3: Feature configurations used to train machine learning models.

order to separately assess the ability of models to handle data from multiple cells and their ability to generalise more widely.

- **'All cells' data split:** The entire dataset was shuffled and then split 60:20:20 between the training, validation and test sets.
- **'Held out cells' data splits:** To assess wider generalisation, all data from five of the seven cells were used for training, while a different cell was used for validation and the remaining cell for testing. The dataset was 'folded' in order to allow every cell to act as test cell once, and also as validation cell once. This resulted in seven dataset folds (Table 4). Importantly, the data within each fold was shuffled to remove possible sources of bias due the sequence of measurements or the ordering of the cells in the dataset.

Feature vectors were standardised prior to model training, i.e. they were scaled according to the mean and variance of each feature in the training data. For the folded

Table 4: Dataset splits



cases, where the training data was different for each fold, the scaling was fold-specific. Scaling transformed each feature to have zero mean and unit variance in the training dataset. The validation and test datasets were scaled according to the scaling parameters computed on their respective training sets. The same scaling strategy (feature-wise scaling) was also applied in the case of spectrograms (config 'G'), although their features are organised in arrays rather than vectors.

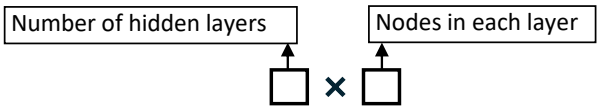
3.2. Models

3.2.1. Regression

The primary model type used with feature configs 'B' to 'F' was the feedforward neural network. FNNs have a structure of fully connected layers, also known as 'dense layers'. Every node in a dense layer has individual connections to all nodes in the previous layer. The number of tuneable parameters in a FNN are defined upon model instantiation, i.e. it is independent of the training process. The more parameters a model has, the greater its 'capacity', meaning that it is capable of fitting a larger set of functions [28]. Nevertheless, high-capacity models are prone to overfitting and require a regularisation strategy. Overfitting is the learning of specific data sequences or dependencies rather than general patterns. Two regularisation strategies were used in this work, the main one being 'early stopping'. An additional set of results was produced by combining early stopping and 'dropout'.

Early stopping is the termination of the training process based on a model's performance on the validation dataset. This is the sole purpose of the validation dataset in our study. A patience of 500 epochs was used in all cases, meaning that model training stopped once the validation loss had not improved for that long. The best version of the model parameters, based on the validation loss, was then restored. An upper limit of 8000 epochs was set, to avoid perpetual training due to marginal or random improvements of the validation loss. Dropout is defined as the random and temporary 'switching off' of neural network nodes at the start of each training epoch. The effect of dropout is to make each hidden node more robust and drive it towards creating useful features on its own, without relying on other nodes to correct its mistakes [29]. Essential concepts such as model capacity and regularisation in artificial neural networks are thoroughly explained in the textbook of Goodfellow, Bengio and Courville [28].

To assess the accuracy of models with respect to their capacity, models of different sizes were constructed and tested with each feature configuration. The number of model parameters in each case is listed in Table 5. The following notation is used in the table to describe FNN models:



Networks of various depths, specifically with 1, 2, and 3 hidden layers, were tested. This is motivated by prior evidence that deep networks can represent certain families of functions more efficiently (more compactly) than shallow networks [30]. In other words, model depth can also contribute to model capacity. Layers with 10, 20, 50, 75, and 100 nodes were constructed, and in deep networks the number of nodes was kept the same for all their layers. FNN models were not trained with feature config 'A' as it captures a very limited feature space that would not justify the model complexity.

Performance benchmarks were set by two classical machine learning methods, using config 'A' as well as configs 'B' to 'F'. The first benchmark is Linear Regression; a prototypically simple model. The second benchmark is the SVM; chosen because of its use by Davies et al [20] who claim a generalised state estimator. It should be noted that in SVMs the concept of model parameters is different from neural networks, and it is not defined at model instantiation. Instead, the number of learned parameters is determined during model training. For this reason, Table 5 does not list the number of parameters for SVMs, but those will be shown in the results section for individual models.

Table 5: Number of model parameters for models used with feature configs A–F.

		Feature configuration					
		A	B	C	D	E	F
Baselines	Linear Regr.	3	10	19	3243	301	3543
	SVM	Training-dependent for SVM					
1 hidden layer	FNN 1 × 10	N/A	111	201	32,441	3,021	35,441
	FNN 1 × 20	N/A	221	401	64,881	6,041	70,881
	FNN 1 × 50	N/A	551	1,001	162,201	15,101	177,201
	FNN 1 × 75	N/A	826	1,501	243,301	22,651	265,801
	FNN 1 × 100	N/A	1,101	2,001	324,401	30,201	354,401
2 hidden layers	FNN 2 × 10	N/A	221	311	32,551	3,131	35,551
	FNN 2 × 20	N/A	641	821	65,301	6,461	71,301
	FNN 2 × 50	N/A	3,101	3,551	164,751	17,651	179,751
	FNN 2 × 75	N/A	6,526	7,201	249,001	28,351	271,501
	FNN 2 × 100	N/A	11,201	12,101	334,501	40,301	364,501
3 hidden layers	FNN 3 × 10	N/A	331	421	32,661	3,241	35,661
	FNN 3 × 20	N/A	1,061	1,241	65,721	6,881	71,721
	FNN 3 × 50	N/A	5,651	6,101	167,301	20,201	182,301
	FNN 3 × 75	N/A	12,226	12,901	254,701	34,051	277,201
	FNN 3 × 100	N/A	21,301	22,201	344,601	50,401	374,601

Feature config 'G' (spectrograms) was used to train CNNs, which are widely used with images. CNNs are capable of identifying structures in two-dimensional datasets by performing transformations to their inputs according to filtering operations. Filters are traversed over the 2D arrays performing a convolution operation, hence the name. Fig. 4 demonstrates at a high level the CNN structure used in this study, which includes two convolutional layers, each succeeded by a max pooling layer. Filters were set to be 3×3 arrays, and these are model parameters that are learned during the training process.

An attribute of CNNs which is particularly appealing is their ability to exhibit a level of positional invariance, i.e. patterns in the 2D dataset can be identified regardless of their exact positions. This is the direct effect of learning the filters after their global application

to a whole image or array. The final stage of the customised CNN architectures, following the convolution and pooling layers, is a series of dense layers. These receive a flattened vector of the filter values as their input, and produce the voltage estimation at the output. A total of 18 CNN architectures were tested in this study, having different sizes, and are listed in Table 6.

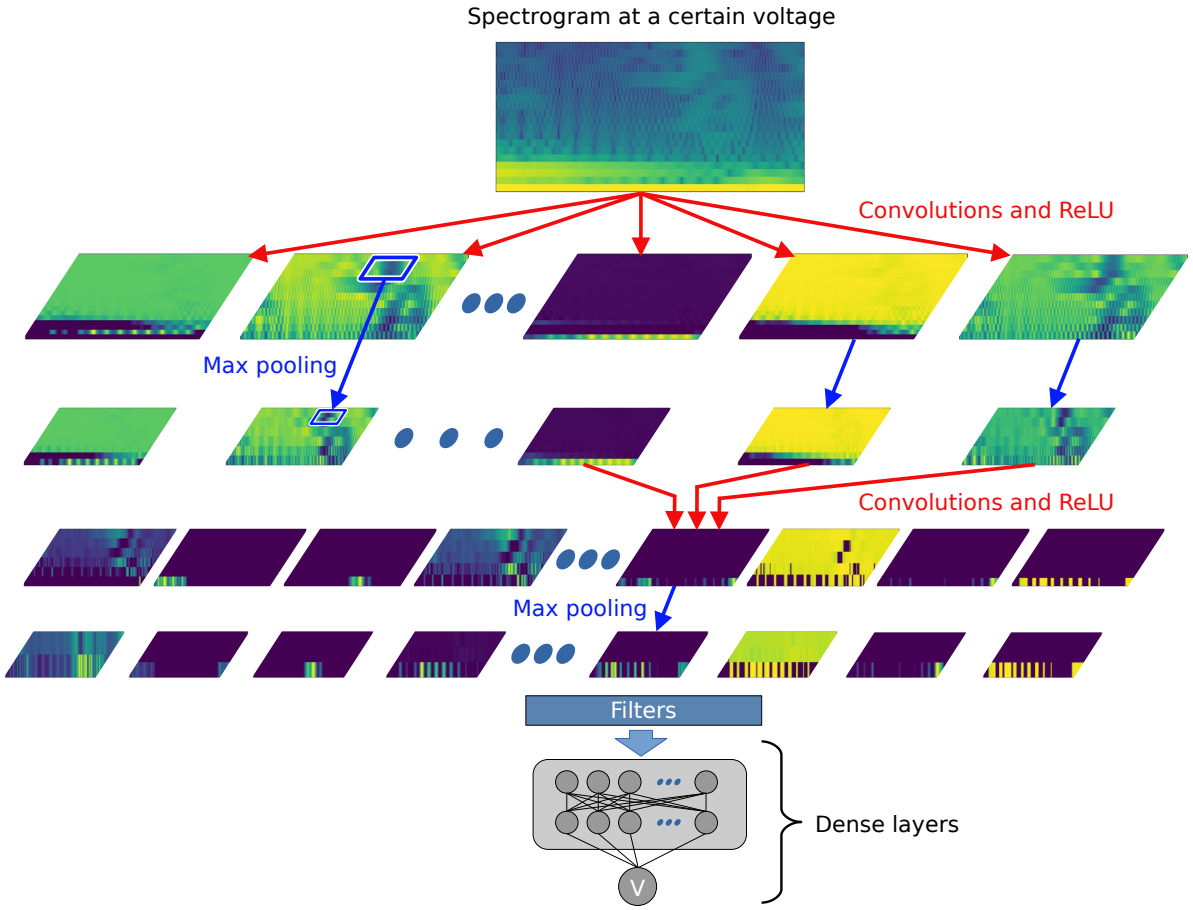


Figure 4: CNN information flow, showing the outputs (not the filters) of each layer (horizontally). Diagram inspired by LeCun, Bengio and Hinton [31].

FNN and CNN models were created and trained using the Tensorflow Python library. The Adam optimiser was used and the learning rate was equal to 0.001 for the first 300 epochs, beyond which point it was set to decay linearly, to reach one-fifth of the starting value on epoch 8000. In cases where dropout was applied, the starting learning rate was set to 0.0005 because this produced more stable training (with less noisy fluctuations of the training and validation losses over progressive epochs). The ReLU activation function was used in all layers of regression models, including the convolutional layers of CNNs. This excludes the output layer which produces the voltage estimation, where no activation was applied. The loss function was the mean absolute error (MAE). SVM and Linear regression models were created and trained using the scikit-learn Python library. In SVM models the default settings of scikit-learn were used, including the radial basis function kernel.

A total of 75 FNN models are listed in Table 5 and 18 CNN models in Table 6. As discussed, all neural networks were trained using the 'all cells' data split and also using

Table 6: Number of model parameters for CNN models used with feature config 'G' (spectrograms).

		Filters in first conv. layer	Filters in second conv. layer	Final dense layers	Num of parameters
1 hidden layer in dense stage	CNN 1	8	16	1 × 50	325,349
	CNN 2	16	32	1 × 50	652,901
	CNN 3	32	64	1 × 50	652,901
	CNN 4	8	16	1 × 100	649,449
	CNN 5	16	32	1 × 100	1,301,001
	CNN 6	32	64	1 × 100	2,611,017
2 hidden layers in dense stage	CNN 7	8	16	2 × 50	327,899
	CNN 8	16	32	2 × 50	655,451
	CNN 9	32	64	2 × 50	1,317,467
	CNN 10	8	16	2 × 100	659,549
	CNN 11	16	32	2 × 100	1,311,101
	CNN 12	32	64	2 × 100	2,621,117
3 hidden layers in dense stage	CNN 13	8	16	3 × 50	330,449
	CNN 14	16	32	3 × 50	658,001
	CNN 15	32	64	3 × 50	1,320,017
	CNN 16	8	16	3 × 100	669,649
	CNN 17	16	32	3 × 100	1,321,201
	CNN 18	32	64	3 × 100	2,631,217

the 7 'held out cells' data folds. Additionally, all models were trained with and without dropout. Therefore, a total of 1488 neural network regression models were created, and their test scores will be discussed in the results section. The implementation of dropout was such that 20% of the nodes in each dense layer were randomly switched off at the start of each training epoch. Dropout was not applied to the input and output layers, or to convolutional layers. All training was carried out on UCL's Myriad High Performance Cluster using Graphics Processing Units (GPUs). Each model ran on a single GPU, i.e. no parallelisation was set up. However, multiple GPUs were used to run models concurrently. The cumulative run time for all regression models was approximately 55 days.

We note that the models created have a broad range of capacities, from 3 parameters to approximately 2.6 million parameters (Tables 5 and 6). Used in combination with the seven feature configurations, it is believed that the performance of these models will be highly indicative of the possibilities and limitations of state estimation using acoustic signals.

3.2.2. Classification

Classification models were trained using the 'all cells' data split for the purpose of recognising the identity of the cell which produced an acoustic waveform. A small number of simple FNN models, containing a relatively small number of parameters, were trained and tested (Table 7). The same training settings and termination strategy as in the regression case were used. Activation functions were again of the ReLU type, except in the output layer where no activation was used. A softmax operation was applied to the model outputs at the point of inference to convert logit vectors to probabilities (logits are the raw, non-normalized predictions). The sparse categorical cross-entropy loss function was used, therefore, labels representing the cell identity were the integers 0 to 6.

Table 7: Number of parameters for FNN classification models.

		Feature configuration			
	FNN	A	B	C	D
1 hidden layer	1 × 2	27	41	59	6,507
	1 × 5	57	92	137	16,257
	1 × 10	107	177	267	32,507

3.2.3. Unsupervised learning

Unsupervised learning models were trained for the purpose of reducing the dimensionality of the different feature configurations to a two-dimensional latent space. This aims to examine whether clusters emerge in the latent space, and if they do, to understand their characteristics. The following techniques were used:

- Principal Component analysis (PCA).
- t- Stochastic Neighbour Embeddings (t-SNE).
- Autoencoders with 1D input-output.
- A Convolutional Autoencoder (CAE) with 2D input-output for use with spectrograms.

PCA is a prototypical linear dimensionality reduction technique with a closed-form solution (Bishop [32], chapter 12). t-SNE is a non-linear, iterative method which is successful at maintaining local structure, meaning that points which are close together in high dimensional space remain close in the lower dimensional projection [33, 34]. Global structure is not necessarily preserved, and points that were far apart in high dimensional space may be brought closer together in projected space. An advantage of this is that the projections tend to be compact and easy to visualise. PCA and t-SNE were applied to feature configs 'B' to 'G', where config 'G' was first flattened into a 1D vector. Data was shuffled to remove order-related biases. PCA and t-SNE were implemented using scikit-learn.

Autoencoders are symmetrical neural networks which progressively compress their input down to a bottleneck, before expanding again to the starting dimensions. During training, an autoencoder aims to reproduce the output from the input and, in this process, it learns a compressed data representation at the bottleneck. Post-training, the structure up to the bottleneck, called the encoder, can be isolated to perform dimensionality reduction by forward pass [28]. Autoencoders were only applied to the richer configurations, 'D', 'F' and 'G' (flattened) using an architecture of dense layers, and config 'G' specifically was also used in its 2D form to train a convolutional autoencoder (Table 8). The number of parameters of all autoencoders is shown in Table 9.

Table 8: Encoder part of the two autoencoder architectures. The decoder is the symmetric expansion of the encoder. Left to right indicates moving from the input layer towards the bottleneck. The symbol † marks layers after which dropout was applied. Dropout was also applied to the input layer itself as a way of introducing noise.

				Nodes in dense layers								
Autoencoder (1D input)	No conv layers			1024 †	512 †	256 †	128	64	32	16	8	2
CAE (2D input)	32 filters	16 filters	8 filters									
	Number of 3×3 filters in conv layers											

Table 9: Number of autoencoder parameters.

Feature configuration	Autoencoder (1D)	Convolutional autoencoder
D	4,726,204	N/A
F	5,033,704	N/A
G	12,677,110	126,026,899

Caution is required when training autoencoders, as given enough model capacity it is possible for them to just learn the identity function between the input and output layer, i.e. to predict each feature of the output layer from the same feature in the input layer [35]. This is a trivial mapping that would not reveal anything useful about the nature of the dataset. Regularisation in autoencoders aims to prevent this type of overfitting; however, the effectiveness of various regularisation strategies differs from other types of neural networks. Vincent et al. [36]. showed that forcing an autoencoder to perform a denoising task as part of the reconstruction process can lead to better learned-representations. Noise can be added to the input layer in various forms. In this study we introduced noise by applying dropout to the input layer at a rate of 20%. This is equivalent to the addition of masking noise to the dataset, which was discussed by Vincent et al. Dropout was also applied to three additional layers of the encoder and decoder as shown in Table 8.

Autoencoders were implemented using Tensorflow. ReLU activations were used on most layers, except for the bottleneck and output layers. A linear activation was applied to the bottleneck to avoid the possibility of one of its two latent dimensions being rectified to zero. A sigmoid activation was applied to the output layer, which we believe can aid the training process given the bounded nature of the waveform amplitude. For compatibility with the sigmoidal output, the input data was first scaled to the [0,1] range.

The training data from the 'all cells' data split was used to train all models. In the case of autoencoders, the validation set was used for early stopping, providing additional regularisation. After training, the test set was used to produce the 2D projections that will be discussed in the results section. Visualising projections on the test set, instead of the whole dataset, provides additional reassurance that any patterns that emerge are not the result of overfitting to the training data. It should be clarified, however, that it does not in itself guarantee that the autoencoders have not simply learned the identity function. The aforementioned denoising strategy is what ameliorated this problem.

4. Results

4.1. Regression

The performance of all regression models at the voltage prediction task, on test data, is shown in Fig. 5. The left and right columns of the figure stand in contrast, where the former represents models trained and tested on data from 'all cells', and the latter represents models trained and tested on folds of 'held out cells'. The results, as a whole, indicate that no model was able to identify patterns which would correlate acoustic signals to the voltage in a way that generalises to the cell population. Models trained on 'all cells' can be successful; however, this must be due to the learning of cell-specific patterns. Specific aspects of the results are discussed next, which reinforce this argument.

Considering the 'all cells' data split (Fig. 5a–c), it is evident that neural networks (FNNs and CNNs) outperformed classical models. In the absence of dropout, this is

true in all cases (Fig. 5b); and it is also true in most cases using dropout (Fig. 5c). Dropout has the effect of decreasing a model's capacity by encouraging the learning of smoother functions. This smoothness constraint likely inhibits the learning of cell-specific patterns, which is a form of overfitting, and would explain the higher errors in Fig. 5(c) compared to (b). In the absence of dropout, increasing the number of parameters or the depth of the networks produced lower errors. Hashed symbols represent a depth of 3 hidden layers, empty symbols a depth of 2 and filled symbols a depth of 1. It can be seen that for each feature configuration, deeper networks outperformed shallow networks with a similar number of parameters. This was not the case when using dropout; depth did not consistently reduce the test error, and the benefit from increasing the model capacity was limited. Both these effects are believed to be due to the regularising effect of dropout, which limited the learning of cell-specific patterns but without equivalent success in promoting the identification of more generic patterns.

Considering the 'held out cells' data splits (Fig. 5d–f), seven times more models are shown because of the seven data folds. Neural networks did not outperform classical models in this case. In fact, the lowest error across the board was achieved by linear regression applied to config 'A' — the simplest feature configuration. This benchmark error is equal to 264 mV, which is too large for any practical application, yet all other models performed even worse when averaged across folds. To put this into context, an estimator predicting a constant value of 3.875 V (the average voltage in the dataset) would result in a MAE of 270 mV, only slightly underperforming the benchmark. Increasing the number of model parameters does not yield any benefit in this case. On the contrary, more severe overfitting and extreme errors are observed when using high-capacity FNN models with some of the richer feature configurations. This is similar to the linear regression case when applied to high dimensional data, although linear regression was not regularised by early stopping. The application of dropout does not have any obvious effect in the 'held out' case either, other than slightly limiting the number of extreme errors. This suggests that the smoothness of the learned functions was irrelevant to their performance on the test set, and that the learned functions did not generally capture characteristics of the test sets.

The inability to generalise to 'held out cells' warrants an investigation of the learning process itself. Fig. 6 shows the number of training epochs for models used with different datasets, with and without dropout. It is reiterated that training was terminated by monitoring the validation loss, with a patience of 500 epochs. Therefore, if a model stopped training exactly on epoch 500 it would mean that its performance on the validation set never improved, even though its starting weights were set randomly. If training stopped soon after epoch 500, then the performance on the validation set either improved very little in the early epochs, possibly due to random fluctuations, or it improved to an optimum very rapidly. Fig. 6b shows that models using a 'held out' cell as the validation dataset stopped training very early in most cases, indicating little or no learning. Examples of longer training are rarer, found mostly when dropout was applied, but still did not achieve a good test-set accuracy as discussed. Fig. 6a shows that using a validation set containing waveforms from all cells (i.e. from the 'all cells' data split) extended the training process significantly, and in some instances training termination was due to reaching the maximum of 8000 epochs. The improvements that led to prolonged training in this case were likely due to the continuous learning and fine-tuning of cell-specific patterns. This process was counteracted when using dropout, resulting in shorter training.

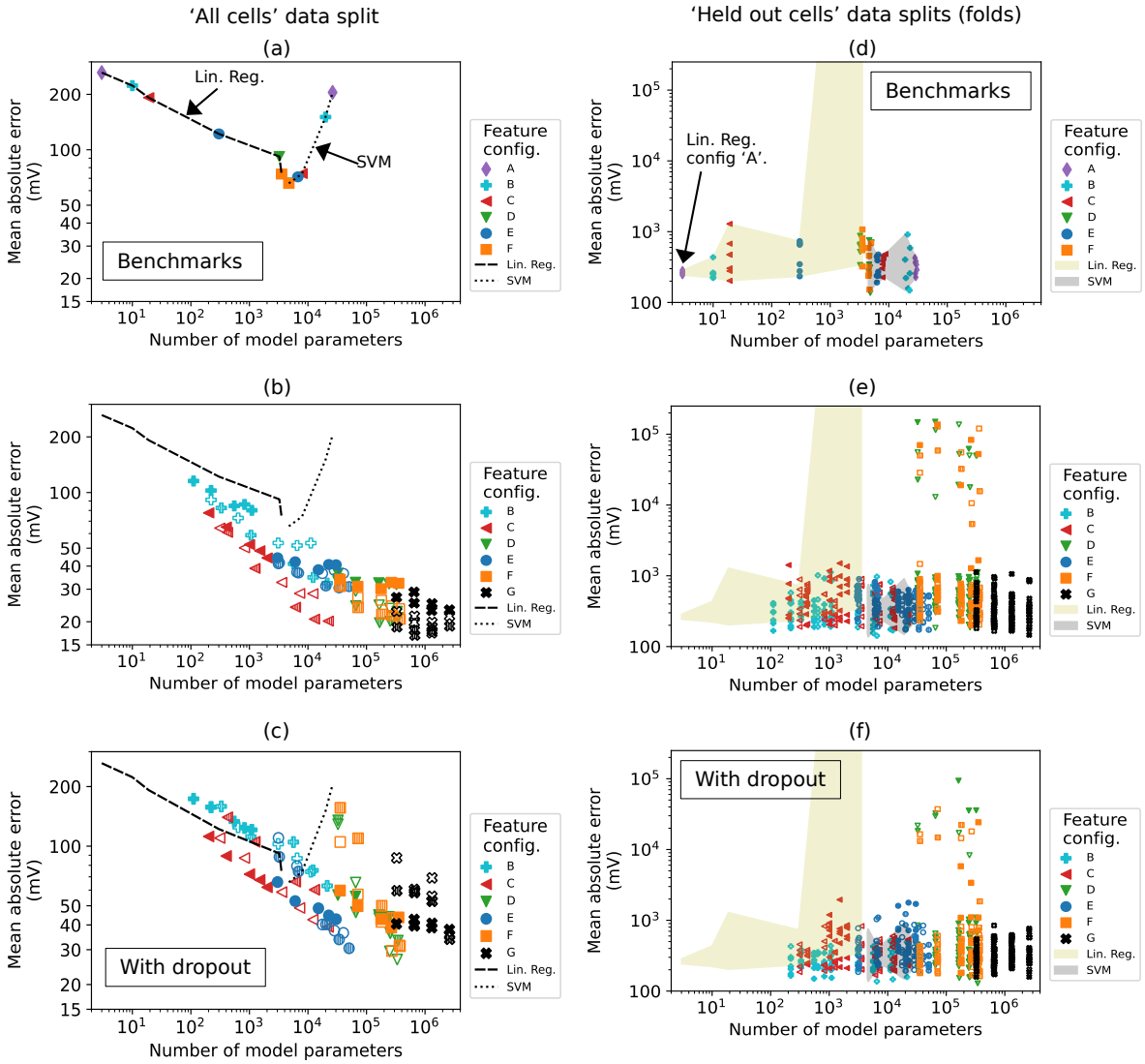


Figure 5: Error values for regression models evaluated on test data. Left column — 'all cells' data split. Right column — 'held out cells' data splits. Filled, empty and hashed symbols are used to distinguish the depth of the neural networks. Filled: 1 hidden layer. Empty: 2 hidden layers. Hashed: 3 hidden layers. Linear Regression and SVM do not have the concept of depth and are shown with filled symbols in the first row, and by lines and shaded regions in the other rows. (a) and (d): Benchmark models. (b) and (e): Models without dropout. (c) and (f): Models with dropout.

4.2. Classification

The accuracy of classification models at predicting the cell identity (7 classes), on test data from 'all cells' (13,399 samples), is shown in Fig. 7. Perfect classification accuracy, equal to 1, was achieved by models with a relatively small number of parameters, even when using simple feature configurations such as 'B' and 'C'. The ease of the classification task suggests that acoustic signals obtained from different cells contain some distinctly different characteristics. This in itself does not guarantee the absence of other acoustic characteristics which could link multiple cells to a common cell state. Nevertheless, it reinforces the hypothesis that the success of regression models on this dataset is likely due to the identification of cell-specific patterns.

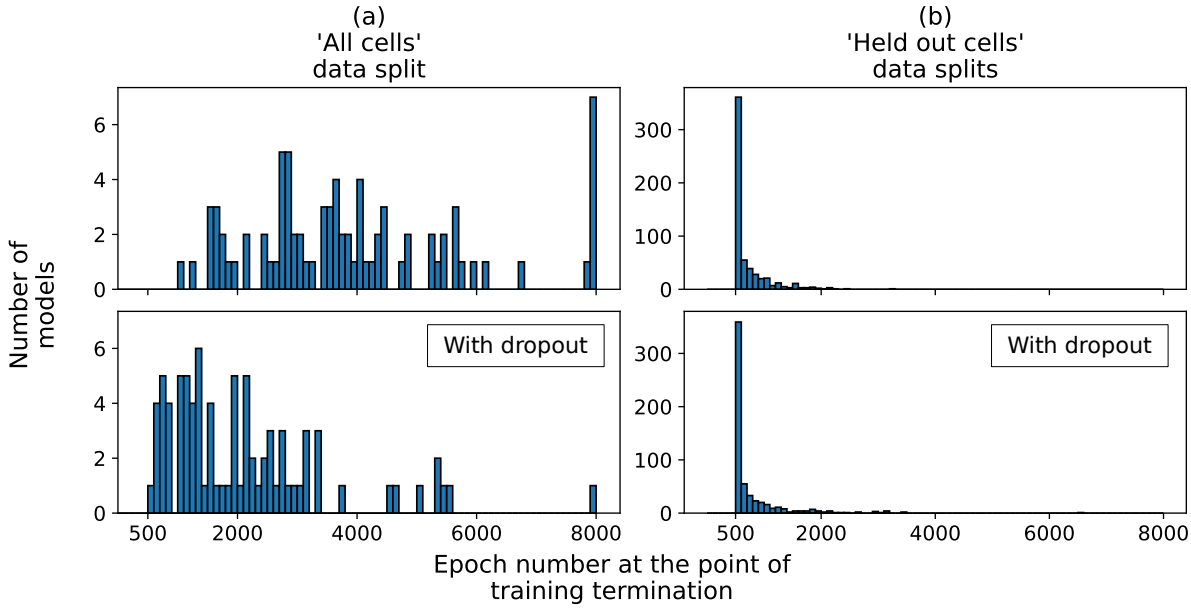


Figure 6: Histograms showing the extent of training for regression FNN and CNN models.

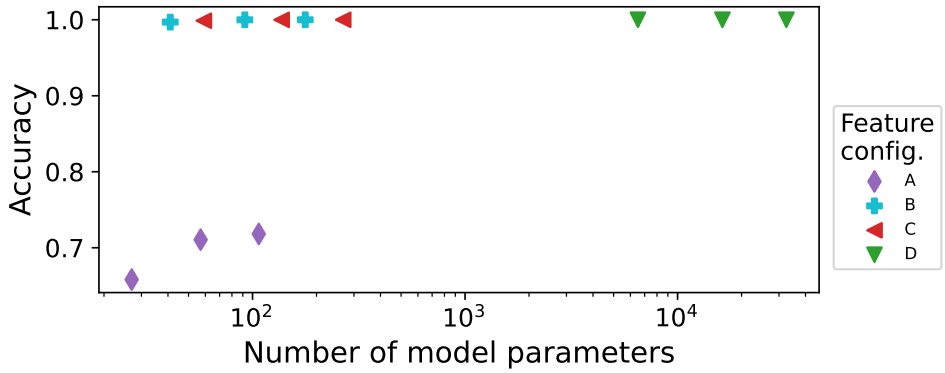


Figure 7: Classification accuracy on test data (13,399 samples). Dataset: 'All cells'.

4.3. Unsupervised Learning

The two-dimensional projections produced by the unsupervised models are shown in Fig. 8. Latent dimensions 1 and 2 are abstract representations generated by the dimensionality reduction techniques that were used. They capture the most significant variance or structural patterns in the data, with each axis representing a combination of the original features that best separates or organizes the data in a lower-dimensional space. The projected waveforms are those of the 'all cells' test set, and were not used in training. Datapoints are coloured according to the cell identity to explicitly examine whether clusters of different cells emerge. It is discovered that this is indeed the case, as individual colours agglomerate together to a large degree in all subplots. This indicates that the distinctiveness of cells in the acoustic dataset dominates over any common links between them and a generic cell state. This distinctiveness appears to be dominant in all models, linear and non-linear, and all feature configurations.

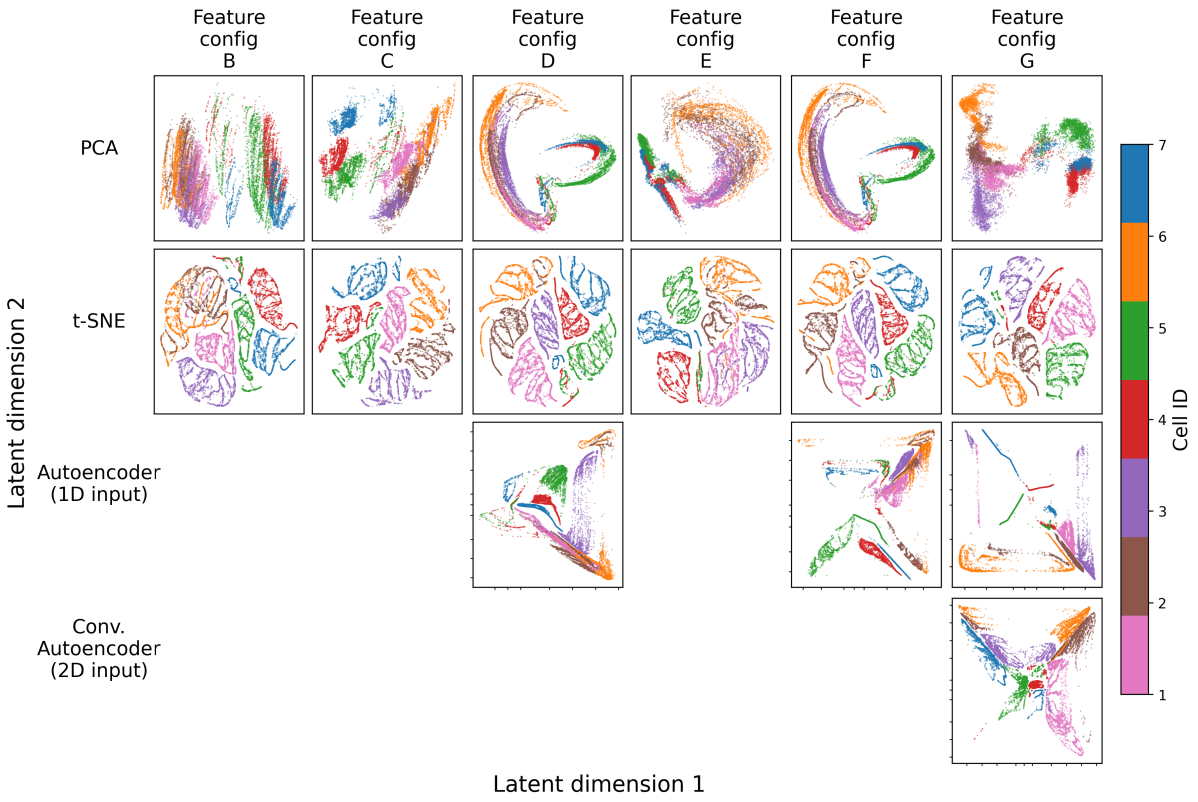


Figure 8: Two-dimensional projections produced by unsupervised learning. Datapoints are coloured by the cell ID. PCA and t-SNE subplots use linear scales. Autoencoders are shown on symlog scales.

5. Discussion

A foundational assumption used in the development of machine learning algorithms is that the samples in a dataset are identically distributed. This means that there exists a data generating probability distribution that can produce all samples of both the training and test sets. The learning task is then to characterize this probability distribution within the structure of a model, allowing useful and accurate predictions to be made on unseen data.

The models used in the regression analysis constitute an extensive search for a data generating distribution that is characteristic of the whole cell population (i.e. characteristic of all cells sold by the manufacturer under a specific catalogue name). Experimentation with a large number of models, having a broad range of model capacities, and using a variety of feature configurations, is what makes this search extensive. The results suggest that such a data generating distribution could not be discovered, and the reasons can be two-fold. Either the dataset itself was insufficient to characterise it; or such a data generating distribution does not exist (meaning that there are no patterns between acoustic features and the SoC that are common across many cells). The latter possibility, which would be an inherent limitation of the acoustic method, should not be readily dismissed. Cell design, manufacturing, and quality control, generally aim to produce cells with consistent electrochemical characteristics. Although this is achieved by carefully controlling the geometry and physical properties of all cell components, the effect of any manufacturing tolerance on the cell's electrochemical characteristics is not necessarily the same as the effect of that same tolerance on the acoustic characteristics. A discrepancy between these

two sensitivities might exist, for example, when considering the thickness of electrode layers, the smoothness of the current collectors and their parallelism, or the amount of excess electrolyte in a cell. Investigating this possible discrepancy between acoustic and electrochemical sensitivities to manufacturing would be useful ground for future research.

In the current study, the lack of generalisation to the cell population may put into question the electrochemical consistency of the cells altogether. Considering the voltage profiles of Fig. 2, a certain degree of cell-to-cell variation is observed. Some of this, however, has been visually exaggerated due to the uncertainty of the x-axis values — the charge level, Q — which experienced drift as discussed. It is reiterated that this uncertainty of the charge level is the reason our study focused on voltage rather than SoC estimation. A complementary investigation of the cell-to-cell voltage variation is provided in SI Section 3, where an additional 10 cells were cycled slowly, at a rate of C/20, to observe their quasi-OCV (qOCV) profiles. The investigation is based on constructing qOCV-versus-SoC curves for the 10 cells and overlaying them, where in this case it was possible to compute the SoC by Coulomb counting because side reactions were minimised thanks to the low C-rate. It was found that between the 10 cells, and at any SoC, the qOCV varied by a maximum of ±31 mV between 10 different cells (SI Fig. 11), and by an average of ±8 mV among all SoCs. It can be concluded that this level of intrinsic cell-to-cell variation does not justify the complete lack of generalisation of the machine learning regression models trained in this chapter, where mean absolute errors were higher than ±264 mV (the baseline case).

It should be acknowledged that a more thorough account of cell-to-cell variation would have also considered the impedimetric profiles of the cells, which affect their overvoltages at the higher C-rates used in this chapter (0.2, 0.5 and 1C). Nevertheless, the additional cell-to-cell discrepancies would need to be very large to hinder generalisation to the extent observed. A more likely explanation for the lack of generalisation is that the acoustic characteristics of the seven cells in the acoustic dataset are distinctly different, as discussed. It is also significant to consider that this distinction persists at all three C-rates — otherwise the classification scores would have been lower, and clusters in the latent spaces produced by unsupervised learning may have been less pronounced. A final consideration, with regard to sources of cell-to-cell variation, is the possibility that the dataset acquired from each cell reflects the local SoC in the tested area, rather than the global SoC measured electrochemically. However, given the small size of the cells tested (*ca.* 4.1 cm^2), spatial SoC variations are likely too minor to account for the lack of model generalisation. Additionally, the transducer used had a contact area of *ca.* 0.3 cm^2, covering a significant portion of each cell (SI Section 4).

Future studies are encouraged to acoustically test a larger population of cells, and to produce population statistics for their acoustic characteristics in comparison to their electrochemical characteristics. Inspiration for the electrochemical characterisation of cell populations can be drawn from Dubarry et al. [37] (100 cells), Rumpf et al. [38] (1100 cells) and Schindler et al. (408 cells). Given the lack of generalisation shown in our study, it is recommended that future datasets also aim to minimise sources of variation. For example, tests at a single slow C-rate are recommended, controlling the cell temperature, and ensuring cell stability. Temperature in the current study varied with a standard deviation of 1.6 °C (max: 26.8 °C, min: 18.7 °C), and three different C-rates were tested. The influence of various C-rates on the acoustic response can be found in our previous work [23]. Another possible direction for future work is to attempt an extension of the acoustic feature space, for example, by performing acoustic frequency sweeps. The identification

of common patterns between cells in the extended feature space, or the lack thereof, can be evaluated by adapting the code [22] and methods put forward in this work.

Lastly, it is worth noting that state estimation on multiple cells using a common model was shown to be possible. MAE close to 15 mV in the voltage estimation task was demonstrated on the seven-cell dataset, and could be further improved by allowing models to train for longer or by fine-tuning their structure and hyperparameters. However, it is critical to recognise that this represents a type of overfitting, where the training process has led to the identification of cell-specific patterns between the acoustic waveforms and the cell state. The implication is that a battery management system using acoustics for state estimation would require every cell to undergo prior testing and to contribute some data for model training. A complementary perspective is to think that the data generating distribution is strongly multimodal, having a different mode per cell. This also explains the ease of the classification task, and the strong presence of clustering-by-cell in reduced dimensions which emerge without supervision.

6. Conclusions

A common pitfall of machine learning applied to battery studies has been demonstrated using an acoustic dataset. It was shown that models with a large number of parameters can have a high enough capacity to capture cell-specific patterns implicitly, without discovering any connecting patterns between cells. This represents a type of overfitting that is frequently disregarded in the literature, and a lack of generalisation which presents a challenge to the wider use of acoustics for battery state estimation. The dataset is the first of its kind to be made available, alongside the trained models [21], and the shared code can be used to reproduce the analysis [22].

The pursuit of generalisation presented is extensive but not exhaustive, and future research can aim to extend the acoustic feature space, and to produce statistics using a larger population of cells under less varied conditions. A shift in focus is recommended, away from demonstrating the correlation of specific acoustic features to states of a single cell, and towards demonstrating the consistency of those features in a population of multiple cells. The machine learning approaches put forward in this work should be useful to this end. Alternative statistical metrics of similarity may also be used.

Acknowledgements

The authors acknowledge the use of the UCL Myriad High Performance Computing Facility (Myriad@UCL), and associated support services, in the completion of this work. The authors also acknowledge the Faraday Institution (Faraday.ac.uk; EP/S003053/1) for the provision of funding as part of the Degradation (FIRG060), Safebatt (FIRG061), Nextrode (FIRG066) and LiSTAR (FIRG058) projects. JBR would also like to thank Innovate UK and the Aerospace Technology Institute for funding through the CEBD programme (10050803). EG acknowledges the EPSRC for funding his studentship through the doctoral training partnership with UCL (EP/N509577/1, EP/T517793/1).

References

[1] B. Sood, M. Osterman, M. Pecht, Health monitoring of lithium-ion batteries, in: 2013 IEEE Symposium on Product Compliance Engineering (ISPCE), 2013, pp. 1–6. doi:10.1109/ISPCE.2013.6664165.

[2] A.G. Hsieh., S. Bhadra, B.J. Hertzberg, P.J. Gjeltema, A. Goy, J.W. Fleischer, D.A. Steingart, Electrochemical-acoustic time of flight: in operando correlation of physical dynamics with battery charge and health, Energy & Environmental Science 8 (5) (2015) 1569–1577. doi:10.1039/C5EE00111K.

[3] G. L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs - Part 3. State and parameter estimation, Journal of Power Sources 134 (2) (2004) 277–292. doi:10.1016/j.jpowsour.2004.02.033.

[4] D. N. T. How, M. A. Hannan, M. S. Hossain Lipu, P. J. Ker, State of Charge Estimation for Lithium-Ion Batteries Using Model-Based and Data-Driven Methods: A Review, IEEE Access 7 (2019) 136116–136136. doi:10.1109/ACCESS.2019.2942213.

[5] L. Gold, T. Bach, W. Virsik, A. Schmitt, J. Müller, T. E. Staab, G. Sextl, Probing lithium-ion batteries' state-of-charge using ultrasonic transmission – Concept and laboratory testing, Journal of Power Sources 343 (2017) 536–544. doi:10.1016/J.JPOWSOUR.2017.01.090.

[6] M. A. Biot, Theory of Propagation of Elastic Waves in a Fluid-Saturated Porous Solid. I. Low-Frequency Range, Journal of the Acoustical Society of America 28 (2) (1956) 168–178. doi:10.1121/1.1908239.

[7] M. A. Biot, Theory of Propagation of Elastic Waves in a Fluid-Saturated Porous Solid. II. Higher Frequency Range, Journal of the Acoustical Society of America 28 (2) (1956) 179–191. doi:10.1121/1.1908241.

[8] Y. Wei, Y. Yan, C. Zhang, K. Meng, C. Xu, State estimation of lithium-ion batteries based on the initial rise time feature of ultrasonic signals, Journal of Power Sources 581 (2023) 233497. doi:10.1016/j.jpowsour.2023.233497.

[9] R. Zhang, X. Li, C. Sun, S. Yang, Y. Tian, J. Tian, State of Charge and Temperature Joint Estimation Based on Ultrasonic Reflection Waves for Lithium-Ion Battery Applications, Batteries 9 (6) (2023). doi:10.3390/batteries9060335.

[10] X. Li, W. Hua, C. Wu, S. Zheng, Y. Tian, J. Tian, State estimation of a lithium-ion battery based on multi-feature indicators of ultrasonic guided waves, Journal of Energy Storage 56 (2022) 106113. doi:10.1016/j.est.2022.106113.

[11] E. Galiounas, T. G. Tranter, R. E. Owen, J. B. Robinson, P. R. Shearing, D. J. Brett, Battery state-of-charge estimation using machine learning analysis of ultrasonic signatures, Energy and AI 10 (2022) 100188. doi:10.1016/J.EGYAI.2022.100188.

[12] S. Montoya-Bedoya, E. Garcia-Tamayo, D. Rohrbach, J. P. Gaviria-Cardona, H. V. Martinez-Tejada, B. Planden, D. A. Howey, W. F. Florez, R. A. Valencia, M. Bernal, Quantitative Ultrasound Spectroscopy for Screening Cylindrical Lithium-Ion Batteries for Second-Life Applications, Batteries & Supercaps n/a (n/a) e202400002. doi:10.1002/batt.202400002.

[13] Y. Liu, R. Zhang, W. Hao, Evaluation of the state of charge of lithium-ion batteries using ultrasonic guided waves and artificial neural network, Ionics 28 (7) (2022) 3277–3288. doi:10.1007/s11581-022-04568-6.

[14] H. Popp, M. Koller, S. Keller, G. Glanz, R. Klambauer, A. Bergmann, State Estimation Approach of Lithium-Ion Batteries by Simplified Ultrasonic Time-of-Flight Measurement, IEEE Access 7 (2019) 170992–171000. doi:10.1109/ACCESS.2019.2955556.

[15] A. Fordham, Z. Milojevic, E. Giles, W. Du, R. E. Owen, S. Michalik, P. A. Chater, P. K. Das, P. S. Attidekou, S. M. Lambert, P. K. Allan, P. R. Slater, P. A. Anderson, R. Jervis, P. R. Shearing, D. J. L. Brett, Correlative non-destructive techniques to investigate aging and orientation effects in automotive Li-ion pouch cells, Joule 7 (11) (2023) 2622–2652. doi:10.1016/j.joule.2023.10.011.

[16] A. S. Leach, A. V. Llewellyn, C. Xu, C. Tan, T. M. M. Heenan, A. Dimitrijevic, K. Kleiner, C. P. Grey, D. J. L. Brett, C. C. Tang, P. R. Shearing, R. Jervis, Spatially Resolved Operando Synchrotron-Based X-Ray Diffraction Measurements of Ni-Rich Cathodes for Li-Ion Batteries, Frontiers in Chemical Engineering 3 (2022). doi:10.3389/fceng.2021.794194.

[17] X. Yu, Z. Feng, Y. Ren, D. Henn, Z. Wu, K. An, B. Wu, C. Fau, C. Li, S. J. Harris, Simultaneous Operando Measurements of the Local Temperature, State of Charge, and Strain inside a Commercial Lithium-Ion Battery Pouch Cell, Journal of The Electrochemical Society 165 (7) (2018) A1578. doi:10.1149/2.1251807jes.

[18] L. Cai, K. An, Z. Feng, C. Liang, S. J. Harris, In-situ observation of inhomogeneous degradation in large format Li-ion cells by neutron diffraction, Journal of Power Sources 236 (2013) 163–168. doi:10.1016/j.jpowsour.2013.02.066.

[19] Z. Huang, Y. Zhou, Z. Deng, K. Huang, M. Xu, Y. Shen, Y. Huang, Precise State-of-Charge Mapping via Deep Learning on Ultrasonic Transmission Signals for Lithium-Ion Batteries, ACS Applied Materials & Interfaces 15 (6) (2023) 8217–8223. doi:10.1021/acsami.2c22210.

[20] G. Davies, K. W. Knehr, B. V. Tassell, T. Hodson, S. Biswas, A. G. Hsieh, D. A. Steingart, State of Charge and State of Health Estimation Using Electrochemical Acoustic Time of Flight Analysis, Journal of The Electrochemical Society 164 (12) (2017) A2746. doi:10.1149/2.1411712JES.

[21] E. Galiounas, R. Jervis, J. Robinson, Dataset: Distinctiveness of acoustic signals from multiple lithium-ion batteries (8 2024). doi:10.5522/04/26843797.v1.

[22] E. Galiounas, SonicBatt: A python package for the visualisation and processing of acoustic signals. https://github.com/EliasGaliounas/SonicBatt. (2024).

[23] E. Galiounas, F. Iacoviello, M. Mirza, L. Rasha, R. E. Owen, J. B. Robinson, R. Jervis, Investigations into the Dynamic Acoustic Response of Lithium-Ion Batteries During Lifetime Testing, Journal of The Electrochemical Society 171 (7) (2024) 70514. doi:10.1149/1945-7111/ad5d21.

[24] R. E. Owen, J. B. Robinson, J. S. Weaving, M. T. M. Pham, T. G. Tranter, T. P. Neville, D. Billson, M. Braglia, R. Stocker, A. A. Tidblad, P. R. Shearing, D. J. L. Brett, Operando Ultrasonic Monitoring of Lithium-Ion Battery Temperature and Behaviour at Different Cycling Rates and under Drive Cycle Conditions, Journal of The Electrochemical Society 169 (4) (2022) 40563. doi:10.1149/1945-7111/ac6833.

[25] W. Dreyer, J. Jamnik, C. Guhlke, R. Huth, J. Moškon, M. Gaberšček, The thermodynamic origin of hysteresis in insertion batteries, Nature Materials 9 (5) (2010) 448–453. doi:10.1038/nmat2730.

[26] V. J. Ovejas, A. Cuadras, Effects of cycling on lithium-ion battery hysteresis and overvoltage, Scientific Reports 9 (1) (2019) 14875. doi:10.1038/s41598-019-51474-5.

[27] K. W. Knehr, T. Hodson, C. Bommier, G. Davies, A. Kim, D. A. Steingart, Understanding Full-Cell Evolution and Non-chemical Electrode Crosstalk of Li-Ion Batteries, Joule 2 (6) (2018) 1146–1159. doi:10.1016/J.JOULE.2018.03.016.

[28] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[30] Y. Bengio, Y. LeCun, Scaling Learning Algorithms toward AI, in: Large-Scale Kernel Machines, The MIT Press, 2007. doi:10.7551/mitpress/7496.003.0016.

[31] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.

[32] C. M. Bishop, Pattern recognition and machine learning, Information science and statistics, Springer, New York, 2006.

[33] L. van der Maaten, G. Hinton, Visualizing data using t-SNE., Journal of machine learning research 9 (11) (2008).

[34] G. E. Hinton, S. Roweis, Stochastic neighbor embedding, Advances in neural information processing systems 15 (2002).

[35] H. Steck, Autoencoders that don t overfit towards the Identity, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 19598–19608.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion., Journal of machine learning research 11 (12) (2010).

[37] M. Dubarry, N. Vuillaume, B. Y. Liaw, From single cell model to battery pack simulation for Li-ion batteries, Journal of power sources 186 (2) (2009) 500–507. doi:10.1016/j.jpowsour.2008.10.051.

[38] K. Rumpf, M. Naumann, A. Jossen, Experimental investigation of parametric cell-to-cell variation and correlation based on 1100 commercial lithium-ion cells, Journal of Energy Storage 14 (2017) 224–243. doi:10.1016/j.est.2017.09.010.