Applied Intelligence (2023) 53:11966–11978
https://doi.org/10.1007/s10489-022-04102-1

Unveil the unseen: Exploit information hidden in noise

Bahdan Zviazhynski[1] · Gareth Conduit[1]

Accepted: 21 August 2022 / Published online: 16 September 2022

Abstract

Noise and uncertainty are usually the enemy of machine learning, noise in training data leads to uncertainty and inaccuracy in the predictions. However, we develop a machine learning architecture that extracts crucial information out of the noise itself to improve the predictions. The phenomenology computes and then utilizes uncertainty in one target variable to predict a second target variable. We apply this formalism to $PbZr_{0.7}Sn_{0.3}O_3$ crystal, using the uncertainty in dielectric constant to extrapolate heat capacity, correctly predicting a phase transition that otherwise cannot be extrapolated. For the second example – single-particle diffraction of droplets – we utilize the particle count together with its uncertainty to extrapolate the ground truth diffraction amplitude, delivering better predictions than when we utilize only the particle count. Our generic formalism enables the exploitation of uncertainty in machine learning, which has a broad range of applications in the physical sciences and beyond.

Keywords Machine learning · Uncertainty · Extrapolation · Case studies

1 Introduction

Throughout the human history, scientific discoveries heralded each new epoch, including the stone, bronze, and iron ages. However, discovering new phenomena is not the only challenge: utilizing the freshly obtained knowledge for real-world applications is crucial. With the availability of computers and large amounts of experimental/computational data nowadays [1–4], machine learning [5–9] has proven an effective tool for this purpose.

Machine learning is a class of methods that start from existing data to train a model and then predict the quantities of interest useful for a given application. For example, machine learning can predict many properties of a putative material [10–18], and moreover can understand the uncertainty in those predictions. This uncertainty can be used to design the material that is most likely to satisfy the set target criteria [19–21], avoiding the typical expensive and time-consuming cycles of trial and improvement experiments. Furthermore, the uncertainty is useful for accelerating materials discovery by guiding where new experiments should be performed in the materials space [22–24], and also for the identification of outliers and erroneous entries in materials databases [25].

While uncertainty is crucial for focusing on the most viable candidates for a given application, uncertainty itself could be a useful value for predicting the quantities of interest. Uncertainty values, computed either analytically or through machine learning, can be used as an input for either analytical or machine learning models that deliver the final predictions. For example, uncertainty (fluctuations) in the financial markets, known as volatility [26], is found analytically and can be hardcoded as an input value for machine learning models to determine prices of derivative contracts [27]. In another example, when an author of a novel deliberately introduces vagueness into a character's speech, the reader infers that the character is unsure about their situation. This uncertainty can be quantified by machine learning models [28, 29] for further use in analytical literary tools. For physical systems, the use of uncertainty for property prediction is motivated by Wilson's Renormalization Group theory [30], in which analytically determined fluctuations on all scales are used in an analytical relationship to determine the macroscopic state

✉ Bahdan Zviazhynski
bz267@cam.ac.uk

Gareth Conduit
gjc29@cam.ac.uk

[1] Theory of Condensed Matter Group, Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, UK

of the system. An example is the liquid-vapor transition, in which near the critical temperature water droplets and vapor intermix over all length scales, leading to critical opalescence [31].

This work develops a methodology that extends previous efforts by using machine learning for both uncertainty estimation and also for the final predictions. This approach utilizes dependences between moments of probability distributions of different quantities, e.g. use uncertainty in one quantity to predict the expected value of another quantity. Moreover, the method can combine the expected value and uncertainty in one quantity, which could increase the amount of information about another quantity and improve the quality of its predictions. An example is shot noise [32], for which the expected value is equal to the square of its uncertainty. Combining both expected value and uncertainty would double the amount of information about the second quantity.

In this paper, we first review the machine learning methods in the literature and set up the formalism to extract information from uncertainty in Section 2. We then validate the formalism on paradigmatic datasets in Section 3 and apply it to the real-world physical examples – $\mathrm{PbZr_{0.7}Sn_{0.3}O_3}$ crystal phase transitions and single-particle diffraction of droplets – in Section 4, predicting the quantity of interest by extrapolation in both cases. Finally, we discuss broader applications of the generic methodology in Section 5.

2 Methodology

We build the individual components for the machine learning methodology before compiling them into a tool to extract information from noise. First, in Section 2.1 we describe the underlying vanilla random forest machine learning method. In Section 2.2 we outline how the machine learning algorithm estimates the uncertainty in its predictions. Following this, in Section 2.3 we address how to handle missing data using an intermediate target variable before finally putting all three components together in Section 2.4 to extract information from uncertainty.

2.1 Machine learning

Machine learning algorithms are trained on existing data to make predictions of target variables for new data entries. A few examples of widely used machine learning algorithms are k-means clustering [33], linear regression [34], neural network [35] and Gaussian processes [36]. In the current work we use random forest, implemented in Scikit-learn Python package [37], since it is computationally cheap, robust against overfitting, and accurate with good uncertainty estimates.

Random forest is a collection of independent identical regression trees [38]. During the training phase, each tree learns the rules for mapping the input variables to target variables. The geometry of regression trees in a random forest, and therefore accuracy of predictions, is affected by the hyperparameters of the random forest. In order to achieve the best accuracy of predictions, we need to select the optimal $min\_samples\_leaf$ hyperparameter, which is the minimum number of datapoints in each leaf of a regression tree. Selection of the optimal hyperparameter – hyperparameter optimization – is done numerically by maximizing the accuracy of a model.

A robust method of assessing the accuracy of a model, applicable to any machine learning algorithm, is k-fold cross-validation. In this method, of the k equally sized subsamples of training data, each one is retained as the validation data for testing the model trained on the remaining subsamples. The process is repeated k times (typically $k = 5$) to obtain the average R^2, coefficient of determination, on validation, which is to be maximized. The cross-validation method is universal so numerically determines the optimal hyperparameters of a machine learning model (e.g. $min\_samples\_leaf$ of a random forest) for any given noisy dataset.

2.2 Uncertainty from machine learning

The sources of uncertainty that machine learning should capture are the inevitable statistical uncertainty in training data derived from experiments, and also the uncertainty in extrapolation. Depending on the machine learning algorithm, different techniques can be employed to estimate the uncertainty in prediction, here we highlight two approaches. Firstly, linear regression [39] and Gaussian processes [40] intrinsically compute the covariance matrix [34] from the training data and use it to estimate the uncertainty.

Secondly, the bootstrap approach can calculate the uncertainty for several methods including neural networks and random forests. More specifically, bootstrap samples of the training set are generated [41]. Each bootstrap sample is obtained by repeatedly drawing an entry from the training set randomly with replacement, meaning any entry can be drawn again in the future, until the bootstrap sample is of the same size as the training set. Then, each bootstrap sample is used to train one machine learning model [19, 35, 42, 43]. The differences among the bootstrap samples lead to differing models that give a range of predictions. The compound predictions are averaged to give the overall prediction, and their standard deviation is the uncertainty in this prediction.

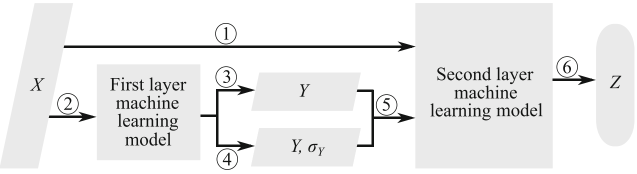

X
①
②
First layer machine learning model
③
④
Y
Y, σ_Y
⑤
Second layer machine learning model
⑥
Z

Fig. 1 Flowchart for the multilayer regressor. Flow 1 takes X directly to the second machine learning model; Flow 2 takes X to the first machine learning model; Flows 3 and 4 predict Y and Y, σ_Y respectively; Flow 5 takes Y and σ_Y to the second layer; Flow 6 outputs Z

2.3 Use of intermediate variable to handle missing data

Data is often erroneous or missing in databases, which limits the information available to train the model and make predictions. We first review the techniques to handle erroneous and missing data in Section 2.3.1. We then describe the method that we use for imputing the missing data in Section 2.3.2.

2.3.1 Techniques to handle erroneous and missing data

Erroneous entries in the data hinder the performance of the model so should be removed. These entries can be pinpointed using outlier detection algorithms, including using Kernel PCA [44], one-class SVM [45], and autoencoders [46]. Another method is to search for entries multiple standard deviations away from the machine learning model prediction [25]. Removing the erroneous entries, however, leads to missing data.

The easiest way to deal with the missing data is to remove the corresponding entries completely [47], but that would lead to loss of crucial information. Another method is to impute the missing values with the mean value of the available data [48], however that would not preserve the relationships between inputs and outputs. Furthermore, the fact that data is missing for a given entry can give crucial information about this entry [49]. For example, if certain clinical measurements are missing from a patient's record, it could either mean that the patient's condition is not severe so the physician saw no need to take the diagnostic test [50], or the condition is severe and therefore the patient dropped out of diagnostic testing to undergo treatment [51].

Machine learning algorithms can learn and exploit correlations between output variables. For example, neural networks that predict different but correlated quantities can share weights to improve predictions [52]. Another example is collaborative filtering [53], which utilizes relationships between online store customer's preferences for different items to give recommendations. Correlations between output variables can be used to impute the missing values [54, 55]. This strategy was generalized to impute the missing data in an iterative and self-consistent manner, and then successfully applied to materials and drugs design [25, 56–58]. Here we first outline this final method in Section 2.3.2, before extending it further in Section 2.4 to exploit the untapped resource of information hidden in noise.

2.3.2 Method to impute missing data

We define a dataset that comprises three columns: an input, an intermediate target variable, and an output. Without loss of generality, we define X as the input, Y as the intermediate target variable, and Z as the output. We can visualize the flow of information through the machine learning algorithm with the flowchart in Fig. 1. A standard machine learning approach will follow Flows 1 & 6 to use only the input feature X to predict the target variable Z.

Instead, we can train the first machine learning model on X (Flow 2) to predict the intermediate target variable Y (Flow 3), which is correlated with the target variable Z. We then train the second machine learning model on Y (Flow 5) to predict Z (Flow 6). The two machine learning models can use different underlying algorithms, making the approach generic and broadly applicable.

Using the intermediate variable Y is particularly useful if we want to extrapolate Z for X outside the training range, e.g. when we have data available for Z at $X < 0$ but no data for Z at $X > 0$, as in the third plot in Fig. 2. At both $X < 0$ and $X > 0$, data is available for Y, as illustrated in the $X-Y$ plot in Fig. 2. We start by learning the $X - Y$ relationship, and then exploit the correlation between Y and Z to improve Z-predictions. In other words, two interpolations $X \rightarrow Y$

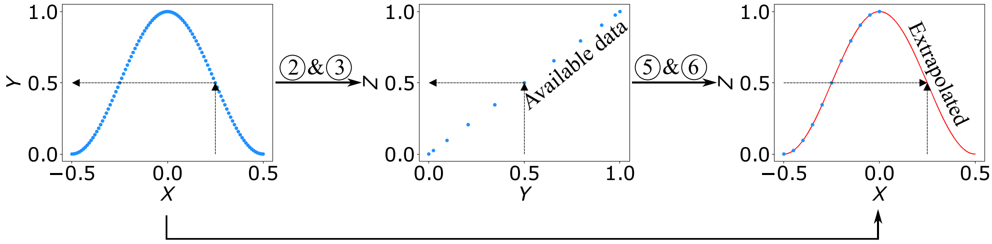


Fig. 2 The three plots applied sequentially utilize variable Y to extrapolate Z on X. In the rightmost figure Z-values are missing for $X > 0$. Blue points are training data, red curve is extrapolated values. The arrows represent Flows 2 & 3 and 5 & 6 from Fig. 1

(Flow 3) and $Y \rightarrow Z$ (Flow 6) would give an extrapolation $X \rightarrow Z$. This situation commonly arises if Z is more expensive to measure than Y and/or more data is available for Y than for Z.

The strategy to exploit an intermediate variable is generic and can be applied to any dataset with correlations between variables. Therefore, the method can be applied to any permutation of input, intermediate, and output columns. For example, previously we discussed $X \rightarrow Y \rightarrow Z$, however the method works just as well to predict $X \rightarrow Z \rightarrow Y$ or $Z \rightarrow Y \rightarrow X$ simply by swapping the column labels in the algorithm. The reordering of the columns can be done concurrently, so within the macromodel the same column can be used as both an input and also be predicted from other columns, depending on what data is available and the quantity of interest.

2.4 Use of uncertainty as an input

When applying machine learning to sparse data, the scarcity of reliable information will always hinder accuracy. It is essential to exploit all available knowledge. Here we develop the formalism to exploit the untapped resource of information hidden in noise. With the three components – machine learning, estimation of uncertainty, and handling of missing data – in place, we are well-positioned to develop the overarching framework to use uncertainty as an input for machine learning. In this methodology we will utilize uncertainty in one target variable for extrapolation of another final target variable. We prescribe the formalism with a general machine learning model, for which any standard method that estimates uncertainty is available, so the approach has broad applicability.

The simplest way to use the uncertainty in target variable Y for extrapolation of another target variable Z is to use a multilayer regressor (Fig. 1). We first train a model (first layer machine learning model) on X (Flow 2) to predict Y and its uncertainty σ_Y (Flow 4); then train another model (second layer machine learning model that can use a different algorithm compared to the first model) on X (Flow 1), Y and σ_Y (Flow 5) to predict Z (Flow 6).

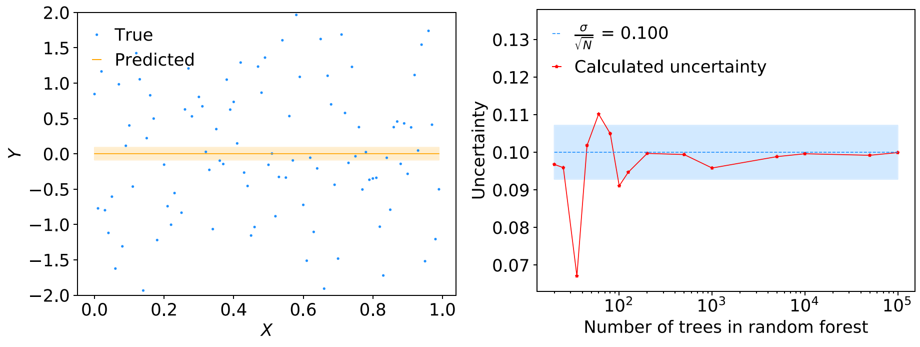
To this point the approach has been prescribed using general and possibly different machine learning models in the two layers. However, we note that as the first and second layer machine learning models predict the same quantity with the same amount of target data, governed by the same underlying trend, they should naturally be similar. Therefore, in practical applications and hereafter in the paper, we set both models to have the same architecture and moreover to adopt the same hyperparameters. This constraint halves the total number of hyperparameters, which not only mitigates overfitting, but moreover reduces the computational cost of hyperparameter optimization.

3 Algorithm validation

Having implemented the machine learning algorithm, we now need to validate its performance. Firstly, in Section 3.1, we test the ability to accurately predict uncertainty. Secondly, in Section 3.2, we confirm the ability to use the prediction of one target variable to predict another target variable. Thirdly, in Section 3.3, we validate the core functionality by estimating the uncertainty in one variable and using it to predict the second variable. Finally, in Section 3.4, we validate the ability of the algorithm to use the combination of one variable and its uncertainty to predict another variable.

3.1 Uncertainty evaluation

Understanding the uncertainty is central to this study, so first we confirm that our machine learning method of choice – random forest – gives a good estimate of the uncertainty in its predictions. We adopt a dataset comprising $N = 100$ entries (Fig. 3a), where $0 < X < 1$ and Y is normally distributed white noise $\sim \mathcal{N}(\mu, \sigma^2)$ with the mean $\mu = 0$ and the variance $\sigma^2 = 1^2$. The hyperparameters that



(a)

(b)

Fig. 3 Random forest for Gaussian white noise: (a) predictions (orange, error region shaded), (b) calculated uncertainty (red) convergence to the true value (blue, error region shaded) as the number of trees is increased

maximize the 5-fold cross-validation R^2 in Y-predictions were found: $min\_samples\_leaf = 100$, leading to each tree averaging over all of the training data, giving a constant prediction equal to the dataset mean, with uncertainty of 0.091 (Fig. 3a). This compares favorably to the analytical estimate of the uncertainty in mean, $\sigma/\sqrt{N} = 0.100 \pm 0.007$ (Fig. 3b, blue dotted line, error region shaded) [59].

When a Gaussian process [40] is applied to the same dataset, it selects a large lengthscale hyperparameter [60] to maximize the 5-fold cross-validation R^2. This averages over the datapoints, similarly to random forest, giving an uncertainty estimate of 0.0995. Therefore, uncertainty estimates from both Gaussian process and random forest are consistent with the analytical value of 0.1. However, since random forest is computationally cheaper, we adopt random forest in this study.

We next check the number of trees, that is number of parallel models trained on different replicas of the data, required to give good estimates of the uncertainty. In Fig. 3b we see that uncertainty estimates using 125 or more trees are all within the error region of $\sigma/\sqrt{N} = 0.100 \pm 0.007$. Therefore, we adopt 125 trees for the remainder of this study.

The choice of $min\_samples\_leaf$ was instrumental to enable the averaging over noise and to give valid uncertainty predictions. Therefore, in order to further investigate the noise averaging lengthscale of random forest, we study a dataset with $0 < X < 10$ and $Y \sim \mathcal{N}(X, 1^2)$ (Fig. 4a). The value of $min\_samples\_leaf$ that minimized the cross-validation MSE in Y-predictions was 25, leading to steps in the predictions, which can be seen in Fig. 4a. Should the random forest average over a higher number of adjacent datapoints, the contribution to the MSE from white noise decreases. However, at the same time, the model underfits the underlying linear function even more. This means that for a given noisy dataset there must be an optimal minimum number of datapoints in the tree leaf, determined numerically through hyperparameter optimization.

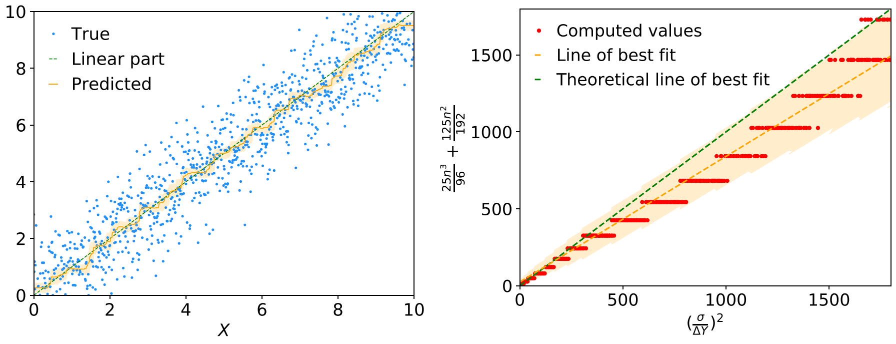
For a more rigorous analysis of the averaging lengthscale, we consider $Y \sim \mathcal{N}(X, \sigma^2)$, and suppose that $n = min\_samples\_leaf$. With points spaced at average increments ΔY on the Y-axis, $n\Delta Y$ is the increase in the underlying linear function across the leaf. This leads to the average contribution to the k-fold cross-validation mean squared error due to underfitting being $\sim \sqrt{\frac{n^2 \Delta Y^2}{12} + \frac{nk^2 \Delta Y^2}{12(k-1)}}$, where $k \ll n$. On the other hand, the contribution to the mean squared error from the noise in the prediction is $\frac{\sigma}{\sqrt{n \frac{k-1}{k}}}$. The two contributions add in quadrature to give the total squared error, which when minimized with respect to n gives:

$$\frac{n^3 k^2 \Delta Y^2}{6(k-1)^2} + \frac{n^2 k^3 \Delta Y^2}{12(k-1)^2} = \sigma^2 \tag{1}$$

Performing numerical experiments, we found the optimal n for several values of σ using machine learning hyperparameter optimization with 5-fold cross-validation. The values of $\frac{25n^3}{96} + \frac{125n^2}{192}$ – left-hand side of (1) with $k = 5$ – plotted against $(\frac{\sigma}{\Delta Y})^2$ can be seen in Fig. 4b as red dots. The plot has a series of plateaus, since n is an integer. A straight line fitted on the red dots (orange line) is within one standard deviation of the theoretical line of best fit (slope of 1, through the origin). This confirms that random forest coupled with hyperparameter optimization gives valid uncertainty predictions, allowing uncertainty to be used as a dependable input for machine learning to predict other quantities later in the paper.

3.2 Extrapolation using intermediate target variable

Having validated that our machine learning model delivers reliable estimates of the uncertainty in its predictions, we turn to confirm its capability to utilize target variables for extrapolation, as first developed in Ref. [25] and then exploited in Refs. [56–58]. This requires a dataset that comprises three columns: one feature column X, an intermediate target column $Y(X) = \cos^2(\pi X)$ (containing information to guide extrapolation), and final target column



(a)

(b)

Fig. 4 Random forest model for a linear function with Gaussian white noise: (a) predictions (orange, error region shaded), (b) plot of $\frac{25n^3}{96} + \frac{125n^2}{192}$ against $(\frac{\sigma}{\Delta Y})^2$, red dots are computed values, orange line is the straight line of best fit, blue line is the theoretical line of best fit (slope of 1, through the origin)

$Z(X) = Y(X) = \cos^2(\pi X)$. In the training set at $-1 < X < 0$, both $Y(X)$ and $Z(X)$ columns have data. At $0 < X < 0.5$, the $Y(X)$ column has data but the $Z(X)$ column is blank – the missing data that we seek to extrapolate for validation. The data can be seen in Fig. 5, and the region with missing Z is shaded in grey in Fig. 5c, whereas all the data with a white background is present.

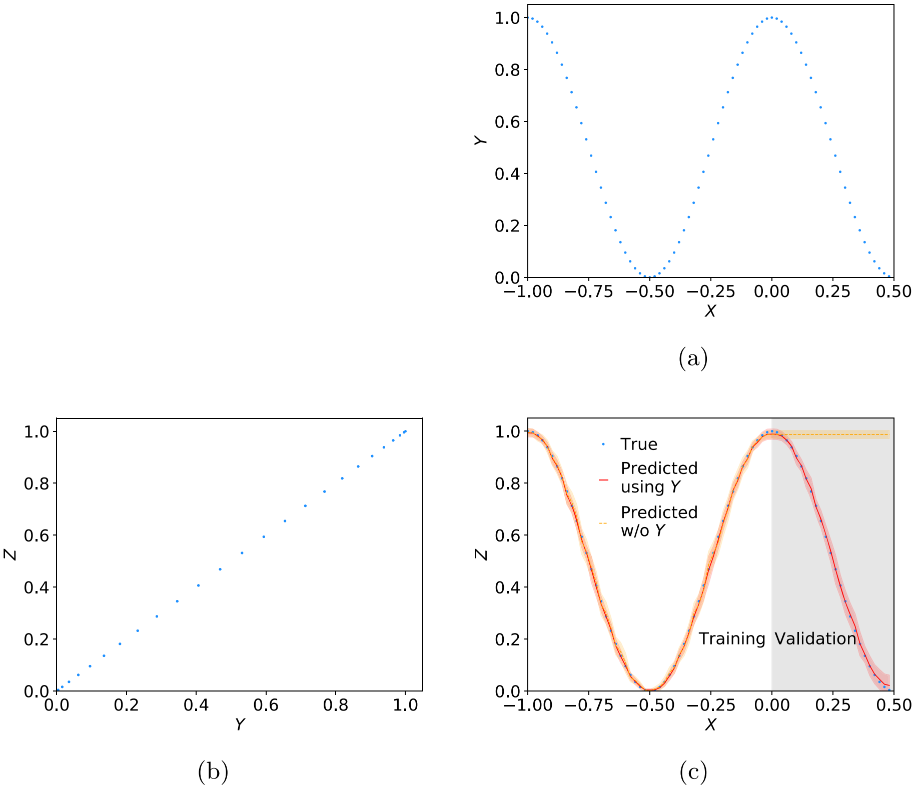
First, working on the training set at $X < 0$, we find the hyperparameters that maximize R^2 in Z-predictions calculated following the blocking cross-validation [61]. In this method, of the three equally sized subsamples of the training set split along X-axis, the two outermost subsamples are retained as validation data for testing the model trained on the remaining subsamples. The model with the tuned hyperparameters, which give the highest R^2 on blocking cross-validation, was trained and then used to predict Z at $X > 0$. The predictions of Z at $X > 0$ were then compared against the true values. There are two strategies to predict $Z(X)$: firstly $X \rightarrow Z$ and secondly $X \rightarrow Y \rightarrow Z$. In Fig. 5, we have a full period of $Z(X)$ $(-1 < X < 0)$ in the training set, so it is more straightforward for the machine learning model to learn the monotonic $Z = Y$ (Fig. 5b) rather than the oscillating $Z(X)$ (Fig. 5c). Therefore, predictions of Z mostly follow the $X \rightarrow Y \rightarrow Z$ strategy. This leads to better predictions of Z at $X > 0$ with $R^2 = 0.9995$ on validation (Fig. 5c). Moreover, this is confirmed through the random forest feature importances of X and Y for predicting Z being 0.003 and 0.997 respectively. When we allow first and second layer models to have different hyperparameters, the prediction accuracy of Z at $X > 0$ drops to $R^2 = 0.997$, which is ascribed to unnecessary freedom in the model leading to overfitting. This demonstrates the importance for both first and second layer models to have identical hyperparameters.

Having seen the good performance of machine learning algorithm to circumvent missing data to extrapolate $Z(X)$, we also confirm that shifting or scaling $Z(X)$ makes no difference to the accuracy, which is expected as random forest is both shift and scale-invariant.

3.3 Extrapolation using uncertainty

We have shown that the multilayer regressor can accurately evaluate the uncertainty and utilize target variables for extrapolation. We now juxtapose these capabilities and validate the algorithm's ability to utilize uncertainty in one variable to extrapolate another variable. We construct a paradigmatic dataset with X, Y, and Z columns, where X is the feature column, $Y \sim \mathcal{N}(0, |Z(X)|^2)$, i.e. the noise is equal to $Z(X)$ and hence $\sigma_Y \propto Z$. Therefore, the second machine learning model in Fig. 1 can learn $X \rightarrow \sigma_Y \rightarrow Z$ (Flows 2, 4, 5 & 6) more easily than $X \rightarrow Z$ (Flows 1 & 6), allowing extrapolation beyond the training range of X. This is analogous to our study of missing data, simply following Flow 4 rather than Flow 3 in the flowchart in

Fig. 5 Prediction of Z using Y for extrapolation with one period in training data. (a) Y vs X on training set (blue), (b) Z vs Y on training set, (c) Z-predictions given X, using Y (red, error region shaded, $R^2 = 0.9995$ on validation) and without Y (orange, error region shaded, $R^2 = -1.75$ on validation). The grey shaded area is the validation set



(a)

(b)

(c)

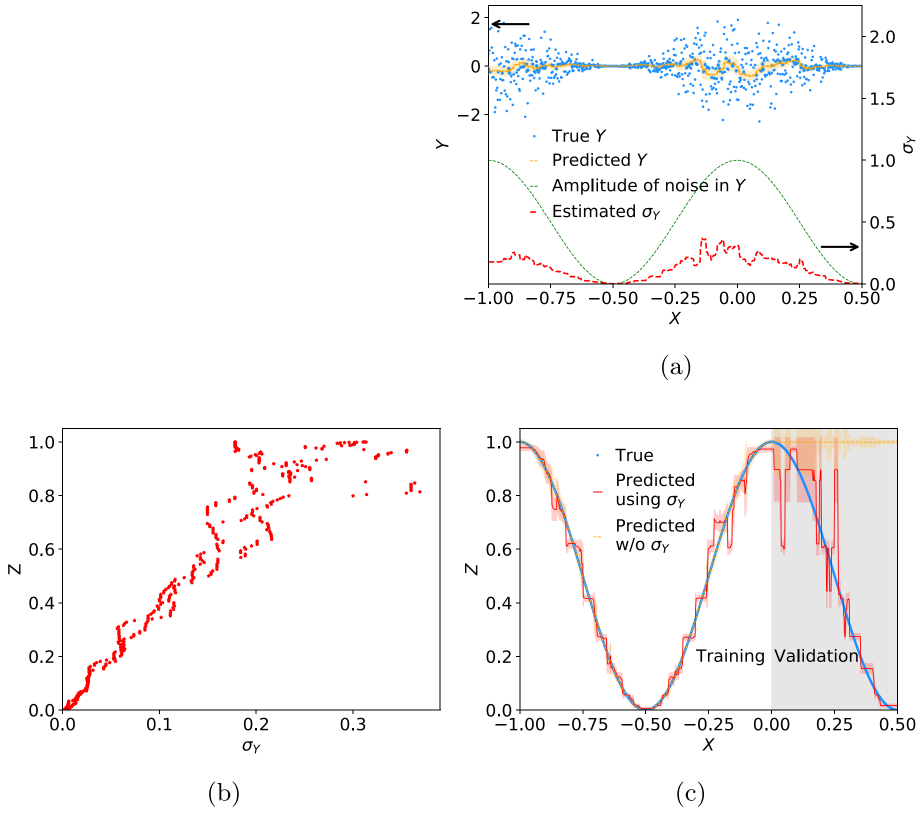
Fig. 1. Therefore, to cement the analogy we adopt the same target function $Z(X) = \cos^2(\pi X)$ as before, with Z-values missing for $X > 0$ for validation.

Following the prescription of our study of missing data, we focus on the case of training on one period ($-1 < X < 0$) of data for $Z(X)$, shown in Fig. 6. We find the hyperparameters that maximize R^2 in Z-predictions calculated following the blocking cross-validation [61]. The optimal value of $min\_samples\_leaf$ was 18, leading to the estimate of the error in predictions, σ_Y, being proportional to the noise in the underlying data, with a proportionality factor of $1/\sqrt{18}$ (Fig. 6a,b). The model with the tuned hyperparameters was trained and then used to predict Z at $X > 0$. These predictions were compared against the true values, giving validation $R^2 = 0.89$ (red curve) and feature importance of σ_Y of 0.93. When we allow first and second layer models to have different hyperparameters, the prediction accuracy of Z at $X > 0$ drops to $R^2 = 0.86$, which is ascribed to overfitting. This again demonstrates the importance of constraining first and second layer models to have identical hyperparameters.

Both first and second layer models can generally predict the same (and often all) target variables (see Section 2.3.2). Initially, the uncertainty in the intermediate target variable, σ_Y, is unknown. Therefore, though the first layer model cannot use the unknown σ_Y as input, it can predict σ_Y through $X \rightarrow \sigma_Y$. This provides a springboard for the second layer model, $\sigma_Y \rightarrow Z$. This means that the two models are not identical. Therefore, they have different internal parameters and hence an increased total number of parameters compared to a single layer model. For clarity, a single layer model cannot learn $\sigma_Y \rightarrow Z$ directly, because no data for σ_Y exists in the initial training data. This demonstrates that in order to exploit the uncertainty for extrapolation, we need more parameters than in a single layer model. Having too many parameters, however, would lead to overfitting on the training set and hence poor predictions on the blind validation set. Our two layer model gives good predictions on the blind validation set (Fig. 6c), and also outperforms a model that does not exploit uncertainty, demonstrating the robustness of the approach.

It should be noted, however, that Z-predictions on validation (Fig. 6c) could have been better – the plateau of Z-predictions extends up to $X = 0.21$, and the minimum value of Z is 0.017 instead of 0. This stems from the noisy estimate of uncertainty in the machine learning predictions of Y, meaning that the model chooses to learn partly from $X \rightarrow Z$. Without using σ_Y, the model mostly learns $X \rightarrow Z$, leading to approximately constant prediction of Z (orange curve).

To study the relative importance of the $X \rightarrow Z$ versus the $X \rightarrow \sigma_Y \rightarrow Z$ approach we now increase the lower bound B of the range $B < X < 0$ of training data and therefore the number of periods of training data at $X < 0$ to increase

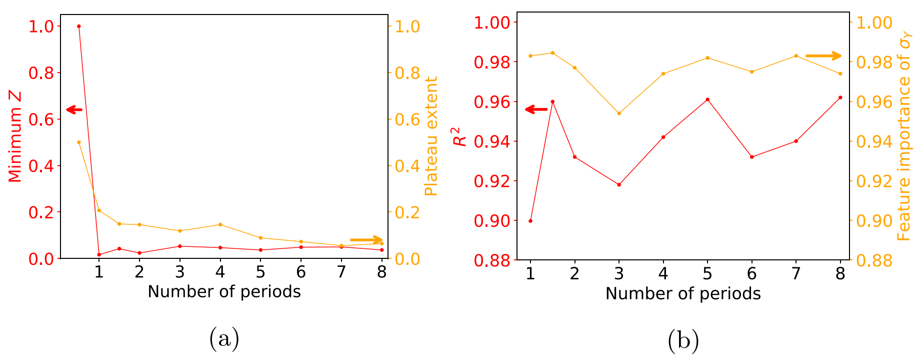


(a)

(b)

(c)

Fig. 6 Predictions from a random forest trained on one period of Gaussian white noise of periodic amplitude: (a) Y (orange, error region shaded) and σ_Y (red) predictions, (b) Z vs σ_Y on training set, (c) Z-predictions given X, using σ_Y (red, error region shaded, $R^2 = 0.89$ on validation) and without σ_Y (orange, error region shaded). The grey shaded area is the validation set



(a)

(b)

Fig. 7 Effect of the number of periods in training set on: (a) Value of Z at the minimum (red) and extent of plateau in Z-predictions (orange); (b) R^2 for validation data (red) and feature importance of σ_Y (orange). Note that at half period, feature importance of σ_Y and R^2 are too low (0.003 and -1.98 respectively) and therefore not shown in the plot

the amount of data to learn the linear $\sigma_Y \rightarrow Z$ relationship. The results can be seen in Fig. 7. With more periods in training set, more information is available for learning the linear $\sigma_Y \rightarrow Z$ relationship, and learning $X \rightarrow Z$ becomes less favourable. This leads to increase in feature importance of σ_Y, consequently improving R^2, minimum Z, and the extent of the plateau on validation. Generally, as little as one period in the training data already gives a real-life benefit from using the uncertainty.

The amplitude of the noise, as long as it is above zero, does not affect the quality of predictions. This is expected, since random forest is scale-invariant. Tests presented so far have been for Gaussian distributed noise. Therefore, the algorithm was tested for other noise distributions including Cauchy, uniform and exponential. For all of these the algorithm delivered predictions with a similar level of accuracy.

3.4 Extrapolation using both intermediate target variable and uncertainty

We have demonstrated that the multilayer regressor can utilize either an intermediate target variable or its uncertainty for extrapolation. If both the intermediate target variable and its uncertainty contain information about the final target variable but are individually noisy, it is possible to combine them to use the less noisy average to help extrapolate the final target variable. This would reduce the mean squared error in predictions of the final target variable by a factor of up to 2. Shot noise [32] is one real-life example of where a variable and its uncertainty are related and so both contain information that we can exploit.

In order to validate the algorithm's ability to combine an intermediate target variable and its uncertainty for extrapolation, we construct a paradigmatic dataset with X, Y, and Z columns. In this dataset, X is the feature column, and $Y \sim \mathcal{N}(Z(X), b^2|Z(X)|^2)$, where b is a positive real number. As before, we adopt $Z(X) = \cos^2(\pi X)$, with one period $(-1 < X < 0)$ of $Z(X)$ for training and Z-values missing for $X > 0$ for validation.

To enable learning of $Z(X)$ from a linear combination of Y and σ_Y, we first standardize Y and σ_Y computed by the first machine learning model $X \rightarrow Y, \sigma_Y$. Then we perform a rotation of Y and σ_Y in the $Y - \sigma_Y$ plane by angle θ:

$$\begin{pmatrix} Y' \\ \sigma'_Y \end{pmatrix} = \begin{pmatrix} \cos(\theta), -\sin(\theta) \\ \sin(\theta), \cos(\theta) \end{pmatrix} \begin{pmatrix} Y \\ \sigma_Y \end{pmatrix} \tag{2}$$

After that, the two rotated components, Y' and σ'_Y, are used as inputs by the second machine learning model to learn Z.

First, working on the training set at $X < 0$, we find the hyperparameters that maximize R^2 in Z-predictions following the blocking cross-validation [61]. The model with the tuned hyperparameters and the optimal angle θ of rotation in the $Y - \sigma_Y$ plane was trained and then used to predict Z at $X > 0$. The predictions are compared against the true values for several values of b, and the results are presented in Fig. 8.

The R^2 values obtained when using rotation (red curve) are slightly better than the R^2 values obtained without using rotation (orange curve) due to the combination of Y and σ_Y being less noisy than the individual quantities. We also compute the R^2 values obtained when the second machine learning model uses only Y (brown curve) and only σ_Y (purple curve). The optimal angle of rotation θ values are

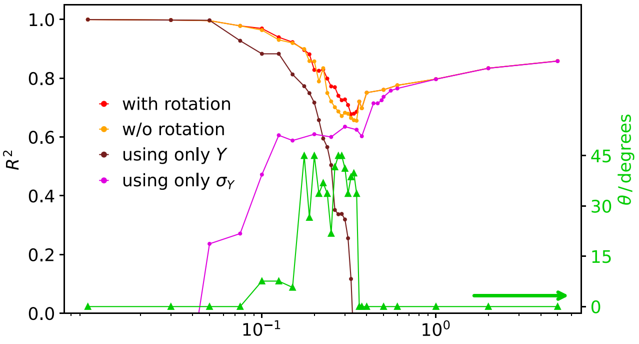


Fig. 8 R^2 for validation data at different b using rotation in $Y - \sigma_Y$ plane (red), without rotation in $Y - \sigma_Y$ plane (orange), using only Y (brown) and using only σ_Y (purple). The green curve shows the optimal angle of rotation θ in $Y - \sigma_Y$ plane

shown by the green curve. It is clear that for the small and large values of b, which correspond to the limits discussed in Sections 3.2 and 3.3, no rotation is necessary due to the fact that Z is predominantly learnt from Y and σ_Y respectively. Both red and orange curves approach the brown curve in the former limit and the purple curve in the latter limit from above.

The noise in σ_Y is smaller than the noise in Y by a factor of $\approx \frac{\sqrt{2}}{b}$, therefore $\theta = 0$ at $b > 0.7$, i.e. it is easier to use only σ_Y for predictions. At at the intermediate values of b less than 0.7, however, θ peaks at 45 degrees, meaning that rotation combines the information from Y and σ_Y in equal proportions to predict Z. Using this rotation marginally improves the predictions (red curve) compared to not using any rotation (orange curve). This improvement comes from averaging the noise in Y and σ_Y, which would otherwise exacerbate predictions of Z in regions where Y and σ_Y are particularly noisy.

4 Application to real-world physical examples

Having set up and validated the machine learning algorithm for extracting information from uncertainty, both directly and alongside the expected value, we are well-positioned to test these two capabilities of the formalism respectively on two real-life physical examples: dielectric crystal phase transitions (Section 4.1) and single-particle diffraction of droplets (Section 4.2). In each case, we take experimental data from the literature and split it into two tranches. We then train the model on the first tranche and validate against the second tranche to replicate a real-life blind prediction against a future experiment.

A general characteristic of a system that machine learning from uncertainty will benefit is non-monotonic $X \rightarrow Z$ behaviour coupled with a noisy intermediate variable Y. An excellent example is a material that undergoes multiple phase transitions as tuning parameter (X) increases. Fluctuations in the system's order parameter (Y) will always be elevated near to each phase transition [62], and so will the energy associated with fluctuations, and therefore heat capacity (Z). Precise measurements of heat capacity are typically performed using differential scanning calorimetry [63], which is usually more costly than measuring order parameter (e.g. dielectric constant). It is therefore attractive to learn $X \rightarrow \sigma_Y$ and then $\sigma_Y \rightarrow Z$, i.e. a map from order parameter fluctuations to heat capacity, in order to extrapolate the latter. In Section 4.1 we apply exactly this approach to a dielectric crystal phase transition.

Systems where combining the value of the intermediate quantity with its uncertainty to achieve statistical averaging are often characterized by counting with shot noise [32]. An example is single-particle diffraction, which we study in Section 4.2. Here, as the diffraction angle (X) varies, the particle count (Y) exhibits noise (σ_Y) that is dependent on Y. The combination of the particle count and its uncertainty can be used to predict the ground truth diffraction pattern (Z), which is obscured in the regions where the particle count itself is finite and therefore noisy.

4.1 $PbZr_{0.7}Sn_{0.3}O_3$ crystal phase transitions

An excellent example of a system that undergoes multiple phase transitions is $PbZr_{0.7}Sn_{0.3}O_3$ – a dielectric crystalline solid. As the temperature (X) increases, this crystal goes from antiferroelectric state (A1) to paraelectric state (PE) via intermediate states A2, IM, and MCC [64] so passes through a total of four phase transitions. The order parameter (Y) is the dielectric constant. The experimental data [64] available for heat capacity already accounts for the Debye contribution [65], leaving the excessive heat capacity (Z) associated with phase transitions, i.e. order parameter fluctuations. The plot of the available experimental data [64] is shown in Fig. 9a.

We focus on first-order phase transitions A1 ↔ A2 and A2 ↔ IM. This pair of first-order transitions are the

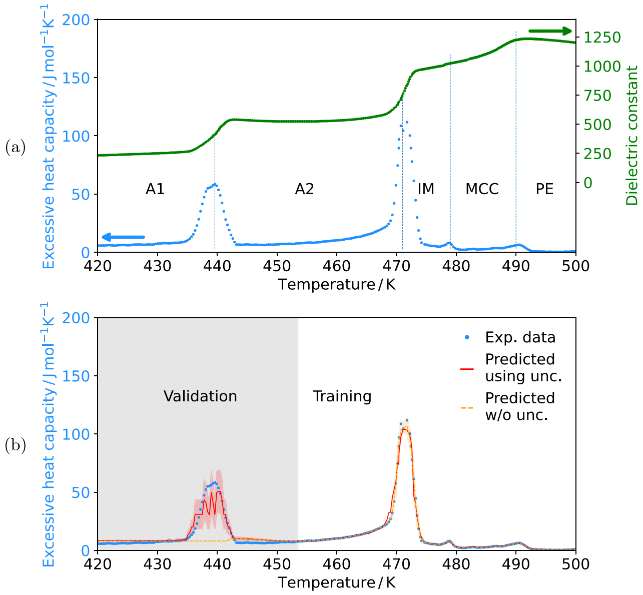
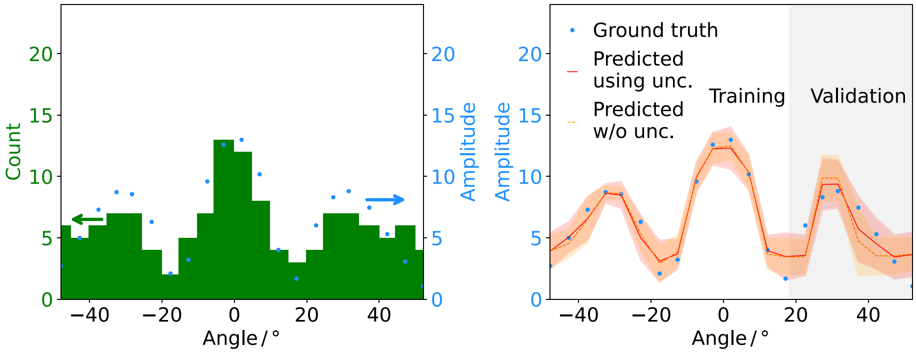


Fig. 9 (a) Experimental data on dielectric constant (green, primary y-axis) and excessive heat capacity (blue, secondary y-axis) against temperature. The vertical black dotted lines denote the phase transitions between the A1, A2, IM, MCC and PE phases. Reproduced from Fig. 5 in Ref. [64]. (b) Predictions of the phase transition at $\sim$ 440 K. The plot includes predictions using uncertainty (red, error region shaded, $R^2 = 0.85$ on validation) and without uncertainty, i.e. only using the mean value of the intermediate variable (orange, error region shaded, $R^2 = -0.04$ on validation). The grey shaded area is the validation region



(a)

(b)

Fig. 10 (a) Data for droplet count (green, primary y-axis) and ground truth amplitude (blue, secondary y-axis) with angle, reproduced from Fig. 3 in Ref. [66]. (b) Predictions of the amplitude for the peak at $\sim$ 30°. The red curve corresponds to predictions using uncertainty ($R^2 = 0.63$ on validation), and the red shaded region is the error region of these predictions. The dashed orange curve corresponds to predictions without using uncertanty as an input, i.e. only using the mean value of the intermediate variable ($R^2 = 0.43$ on validation), and the orange shaded region is the error region of these predictions. Blue points are the original data. The grey shaded area is the validation region

most prominent in Fig. 9a, whereas the other second-order transitions, despite having diverging fluctuations, have a narrow region of sharp increase in heat capacity that is not resolved in the available experimental data.

We train a multilayer regressor on the peak at $\sim$ 471 K and validate on the peak at $\sim$ 440 K. For training, the dielectric constant data is available at all temperatures, but the excessive heat capacity is only available to the right of the first peak, therefore provides no information about the phase transition at $\sim$ 440 K. The results can be seen in Fig. 9b. The algorithm delivers strong predictions for heat capacity when compared to unseen data at temperatures below 455 K with $R^2 = 0.85$, correctly identifying the phase transition at $\sim$ 440 K. The feature importance of uncertainty in dielectric constant is 0.87, showing the utility of uncertainty in understanding the phase transitions. If the machine learning is trained without being able to exploit uncertainty, we see the predictions completely miss the phase transition with $R^2 = -0.04$, confirming the importance of predicting and exploiting the uncertainty.

4.2 Single-particle diffraction of droplets

Having successfully applied our method to dielectric crystal phase transitions, we proceed to demonstrate its applicability to another phenomenon – diffraction of droplets through a double slit, taking data from Ref. [66]. For a given angle (X) the count of droplets (Y) diffracted into that angle was measured and plotted on a histogram. The flow of droplets is low, making accumulation of data for the diffraction pattern time-consuming. We therefore turn to machine learning to take available data and estimate the diffraction pattern, expecting it to be that of a double slit. The noise in the count (σ_Y) is expected to follow a Poisson distribution [67], i.e. to depend on the expected value of the count. Therefore, it is possible to use both Y and σ_Y self-averaging the statistical uncertainty in each, delivering a more precise estimate for the analytical ground truth amplitude (Z), despite the finite number of droplets in the experiment. The determination of the ground truth amplitude is useful for investigating the properties of the particles source (e.g. particle energy) if at some angles the particle count is noisy or low.

Experimental data for diffraction of 75 particles was taken from Ref. [66] and is shown in Fig. 10a. We train multilayer regressor on the two peaks at $\sim$ −30° and $\sim$ 0° and validate on the peak at $\sim$ 30°. For training, the count is available at all angles, but the ground truth amplitude is only available at the peaks at $\sim$ −30° and $\sim$ 0°. Such a situation may arise when the ground truth amplitude is unknown at some angles due to the complex nature of the particles source and/or the aperture. The results can be seen in Fig. 10b. The model achieves $R^2 = 0.63$ on validation. Without using the uncertainty, the value of R^2 on the validation set is 0.43, confirming the significant benefit of the use of uncertainty to improve extrapolation. The mean squared error in predictions is reduced by a factor of $\frac{1-0.43}{1-0.63} = 1.54 < 2$, in agreement with the expected improvement when using uncertainty discussed in Section 3.4.

5 Conclusion

We developed, implemented, and validated a machine learning framework which, given an input feature X, calculates uncertainty in target variable Y, σ_Y, and uses Y and/or σ_Y to predict another target variable Z. Two successive interpolations $X \rightarrow Y, \sigma_Y$ and $Y, \sigma_Y \rightarrow Z$ enable the difficult extrapolation $X \rightarrow Z$. Tests on paradigmatic datasets show two significant advantages: firstly the exploitation of information only in σ_Y, and secondly reduction of noise by averaging Y and σ_Y when they are proportional.

To showcase the method it was applied to two experimental datasets. For the first dataset on dielectric crystal, given the temperature (X) range and the order parameter (Y) values and exploiting the uncertainty σ_Y in Y-predictions, heat capacity (Z) was extrapolated with respect to temperature. The method quantitatively predicted the phase transition completely missed by standard machine learning methods. For the second dataset, single-particle diffraction of droplets, given the angle (X) range and the particle count (Y) values and exploiting the Y-values together with the uncertainty σ_Y in Y-predictions, the ground truth amplitude (Z) was extrapolated with respect to angle. Our method that combines Y and σ_Y improves extrapolation in the region with noisy Y-values, reducing the mean squared error by a factor of ~ 2. This demonstrates the importance of uncertainty as a source of information in its own right, to improve the predictive power of machine learning methods for physical phenomena.

Furthermore, the method can operate on any number of input features and target variables and the generic algorithm can be applied in many different situations. This endorses the method's applicability to more complex physical systems, e.g. concrete or atomic junctions. For concrete, the input feature X would be the position within the image of the material's microstructure. The intermediate target variable Y would be the size/contrast of the aggregate, and the uncertainty in it, σ_Y, is linked to mechanical properties of the material (Z), such as strength [68]. For atomic junctions, the input feature X would be the shape/structure of the junction. The intermediate target variable Y would be the count of electrons passing through the junction. This count has shot noise, hence the combination of Y and σ_Y can be used to improve predictions of properties linked to electron count, such as conductivity (Z) [69].

The algorithm has potential applications in areas beyond physics as well. One of these areas is financial markets, where higher uncertainty in predictions of future stock price movement leads to investors being less likely to buy or sell it, i.e. to decrease in its trading volume [70]. Another example is cancer, which is known to cause genetic chaos [71]. The information extracted from this chaos can be used for early cancer detection.

Acknowledgements The authors acknowledge the financial support of the Engineering and Physical Sciences Research Council, Harding Distinguished Postgraduate Scholars Programme Leverage Scheme, and the Royal Society (Grant No. URF\R\201002). There is Open Access to this paper at https://www.openaccess.cam.ac.uk. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Author Contributions **Bahdan Zviazhynski**: Methodology, Software, Formal analysis, Data curation, Writing – original draft. **Gareth Conduit**: Conceptualisation, Methodology, Writing – review & editing, Supervision.

Declarations

Conflict of Interests Gareth Conduit is a Director of materials machine learning company Intellegens. The authors have no other financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Andersen CW, Armiento R, Blokhin E, Conduit GJ et al (2021) OPTIMADE, an API for exchanging materials data. Nature Scientific Data 8:217. https://doi.org/10.1038/s41597-021-00974-z
2. Granta, Design (2017) CES EduPack. https://www.grantadesign.com/industry/products/data/materialuniverse/
3. NoMaD (2017) https://nomad-lab.eu/index.php?page=repo-arch
4. MatWeb LLC (2017) http://www.matweb.com/
5. Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin
6. Guerney K (1997) An Introduction to Neural Networks. UCL Press
7. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, vol 25. Curran Associates, Inc
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90
9. Karpathy A, Toderici G, Shetty S, Leung T et al (2014) Large-scale video classification with convolutional neural networks. In: 2014 IEEE Conference on computer vision and pattern recognition, 1725–1732. https://doi.org/10.1109/CVPR.2014.223
10. Bhadeshia HKDH, MacKay DJC, Svensson LE (1995) Impact toughness of C-Mn steel arc welds – Bayesian neural network analysis. Mater Sci Technol 11:1046–1051. https://doi.org/10.1179/mst.1995.11.10.1046
11. Sourmail T, Bhadeshia H, MacKay DJC (2002) Neural network model of creep strength of austenitic stainless steels. Mater Sci Technol 18:655–663. https://doi.org/10.1179/026708302225002065

12. Agrawal A, Deshpande PD, Cecen A, Basavarsu GP et al (2014) Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. Integr Mater Manuf Innov 3:1–19. https://doi.org/10.1186/2193-9772-3-8
13. Ward L, Agrawal A, Choudhary A, Wolverton C (2016) A general-purpose machine learning framework for predicting properties of inorganic materials. npj Comput Mater 2:16028. https://doi.org/10.1179/mst.1995.11.10.1046
14. Legrain F, Carrete J, van Roekeghem A, Curtarolo S et al (2017) How Chemical Composition Alone Can Predict Vibrational Free Energies and Entropies of Solids. Chem Mater 29:6220–6227. https://doi.org/10.1021/acs.chemmater.7b00789
15. Gomberg JA, Medford AJ, Kalidindi SR (2017) Extracting knowledge from molecular mechanics simulations of grain boundaries using machine learning. Acta Mater 133:100–108. https://doi.org/10.1016/J.ACTAMAT.2017.05.009
16. Ubaru S, Mikeldar A, Saad Y, Chelikowsky JR (2017) Formation enthalpies for transition metal alloys using machine learning. Phys Rev B 95:214102. https://doi.org/10.1103/PhysRevB.95.214102
17. Lee J, Seko A, Shitara K, Nakayama K et al (2016) Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques. Phys Rev B 93:115104. https://doi.org/10.1103/PhysRevB.93.115104
18. Ward L, Liu R, Krishna A, Hegde VI et al (2017) Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations. Phys Rev B 96:024104. https://doi.org/10.1103/PhysRevB.96.024104
19. Conduit BD, Jones NG, Stone HJ, Conduit GJ (2017) Design of a nickel-base superalloy using a neural network. Mater Design 131:358. https://doi.org/10.1016/j.matdes.2017.06.007
20. Conduit BD, Jones NG, Stone HJ, Conduit GJ (2018) Probabilistic design of a molybdenum-base alloy using a neural network. Scripta Mater 146:82. https://doi.org/10.1016/j.scriptamat.2017.11.008
21. Conduit BD, Illston T, Baker S, Duggappa DV et al (2019) Probabilistic neural network identification of an alloy for direct laser deposition. Mater Design 168:107644. https://doi.org/10.1016/j.matdes.2019.107644
22. Dehghannasiri R, Xue D, Balachandran PV, Yousefi MR et al (2017) Optimal experimental design for materials discovery. Comput Mater Sci 129:311. https://doi.org/10.1016/j.commatsci.2016.11.041
23. Xue D, Balachandran PV, Hogden J, Theiler J et al (2016) Accelerated search for materials with targeted properties by adaptive design. Nature Commun 7:11241. https://doi.org/10.1038/ncomms11241
24. Smith JS, Nebgen B, Lubbers N, Isayev O et al (2018) Less is more: Sampling chemical space with active learning. J Chem Phys 148:241733. https://doi.org/10.1063/1.5023802
25. Verpoort PC, MacDonald P, Conduit GJ (2018) Materials data validation and imputation with an artificial neural network. Comput Mater Sci 147:176. https://doi.org/10.1016/j.commatsci.2018.02.002
26. Daly K (2008) Financial volatility: Issues and measuring techniques. Physica A 387:2377–2393. https://doi.org/10.1016/j.physa.2008.01.009
27. Zhang L (2020) A general framework of derivatives pricing. J Math Financ 10:255–266. https://doi.org/10.4236/jmf.2020.102016
28. Zerva C, Batista-Navarro R, Day P, Ananiadou S (2017) Using uncertainty to link and rank evidence from biomedical literature for model curation. Bioinformatics 33(23):3784–3792. https://doi.org/10.1093/bioinformatics/btx466
29. Goujon B (2009) Uncertainty detection for information extraction. In: Proceedings of the international conference RANLP-2009 association for computational linguistics, Borovets Bulgaria
30. Wilson KG (1983) The renormalization group and critical phenomena. Rev Mod Phys 55:583. https://doi.org/10.1103/RevModPhys.55.583
31. Gopal ESR (2000) Critical opalescence. Resonance 5:37–45. https://doi.org/10.1007/BF02837904
32. Perepelitsa VD (2006) Johnson noise and shot noise. MIT Department of Physics
33. Cohn R, Holm E (2021) Unsupervised machine learning via transfer learning and *k*-means clustering to classify materials image data. Integr Mater Manuf Innov 10:231–244. https://doi.org/10.1007/s40192-021-00205-8
34. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. 2nd edition. Springer
35. Heskes T (1997) Selecting weighting factors in logarithmic opinion pools. In: Advances in neural information processing systems, vol 10. MIT Press
36. Tancret F (2013) Computational thermodynamics, Gaussian processes and genetic algorithms: combined tools to design new alloys. Modelling Simul Mater Sci Eng 21:045013. https://doi.org/10.1088/0965-0393/21/4/045013
37. Pedregosa F, Varoquaux G, Gramfort A, Michel V et al (2011) Scikit-learn: Machine learning in python. J Mach Learn Res 12:2825–2830
38. Loh WY (2011) Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1:14–23. https://doi.org/10.1002/widm.8
39. Bagos P, Adam M (2015) On the covariance of regression coefficients. Open J Stat 05(07):680–701. https://doi.org/10.4236/ojs.2015.57069
40. Williams C, Rasmussen C (1995) Gaussian processes for regression. In: Advances in neural information processing systems, vol 8. MIT Press
41. Efron B (1979) Bootstrap methods: Another look at the jackknife. Ann Statist 7(1):1–26. https://doi.org/10.1214/aos/1176344552
42. Lee TH, Ullah A, Wang R (2020) Bootstrap aggregating and random forest. In: Macroeconomic forecasting in the era of big data. Advanced studies in theoretical and applied econometrics, vol 52. Springer, Cham. https://doi.org/10.1007/978-3-030-31150-6_13
43. Papadopoulos G, Edwards PJ, Murray AF (2001) Confidence estimation methods for neural networks: a practical comparison. IEEE Trans Neural Netw 12(6):1278–1287. https://doi.org/10.1109/72.963764
44. Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319. https://doi.org/10.1162/089976698300017467
45. Schölkopf B, Williamson RC, Robert C, Smola A et al (1999) Support vector method for novelty detection. In: Advances in neural information processing systems, vol 12. MIT Press
46. Borghesi A, Bartolini A, Lombardi M, Milano M et al (2019) Anomaly detection using autoencoders in high performance computing systems. In: Proceedings of the AAAI Conference on artificial intelligence, vol 33, pp 9428–9433
47. Fouad KM, MM MMI, Azar AT, Arafa MM (2021) Advanced methods for missing values imputation based on similarity learning. PeerJ Comput Sci 7:e619. https://doi.org/10.1016/j.neucom.2014.02.037
48. Ravi V, Krishna M (2014) A new online data imputation method based on general regression autoassociative neural network. Neurocomputing 138:106–113. https://doi.org/10.1016/j.neucom.2014.02.037
49. Wells BJ, Chagin KM, Nowacki AS, Kattan MW (2013) Strategies for handling missing data in electronic health

record derived data. EGEMS (Washington, DC) 1(3):1035. https://doi.org/10.13063/2327-9214.1035

50. Groenwold RHH (2020) Informative missingness in electronic health record systems: The curse of knowing. Diagn Progn Res 4:8. https://doi.org/10.1186/s41512-020-00077-0
51. Haneuse S, Arterburn D, Daniels MJ (2021) Assessing missing data assumptions in EHR-Based studies: A complex and underappreciated task. JAMA Netw Open 4(2):e210184. https://doi.org/10.1001/jamanetworkopen.2021.0184
52. Roth W, Pernkopf F (2020) Bayesian neural networks with weight sharing using Dirichlet processes. IEEE Trans Pattern Anal Mach Intell 42(1):246–252. https://doi.org/10.1109/TPAMI.2018.2884905
53. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: The adaptive Web: Methods and strategies of Web personalization, pp 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_9
54. Khan H, Wang X, Liu H (2022) Handling missing data through deep convolutional neural network. Inf Sci 595:278–293. https://doi.org/10.1016/j.ins.2022.02.051
55. Lokupitiya RS, Lokupitiya E, Paustian K (2006) Comparison of missing value imputation methods for crop yield data. Environmetrics 17:339–349. https://doi.org/10.1002/env.773
56. Mahmoud SY, Irwin BWJ, Chekmarev D, Vyas S et al (2021) Imputation of sensory properties using deep learning. J Comput-Aided Mol Des 35:1125. https://doi.org/10.1007/s10822-021-00424-3
57. Irwin BWJ, Levell J, Whitehead TM, Segall MD et al (2020) Practical applications of deep learning to impute heterogeneous drug discovery data. J Chem Inf Model 60:2848. https://doi.org/10.1021/acs.jcim.0c00443
58. Whitehead TM, Irwin BWJ, Hunt PA, Segall MD et al (2019) Imputation of assay bioactivity data using deep learning. J Chem Inf Model 59:1197. https://doi.org/10.1021/acs.jcim.8b00768
59. Johnson NL, Kotz S, Balakrishnan N (1994) Continuous univariate distributions. 2nd edn. Wiley, New York
60. Rasmussen CE, Edwards CKI (2006) Gaussian processes for machine learning. The MIT Press
61. Roberts DR, Bahn V, Ciuti S, Boyce MS et al (2017) Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40:913–929. https://doi.org/10.1111/ecog.02881
62. Cheung A (2011) Phase transitions lecture notes. University of Cambridge, Cambridge
63. Gill P, Moghadam TT, Ranjbar B (2010) Differential scanning calorimetry techniques: applications in biology and nanoscience. J Biomol Tech 21(4):167–193
64. Jankowska-Sumara I, Podgorna M, Majchrowski A, Zukrowski J (2017) Thermal analysis of phase transitions in $\mathrm{PbZr}_{1-x}\mathrm{Sn}_x\mathrm{O}_3$ antiferroelectric single crystals. J Therm Anal Calorim 128:713–719
65. Schliesser JM, Woodfield BF (2015) Development of a Debye heat capacity model for vibrational modes with a gap in the density of states. J Phys: Condens Matter 27:285402
66. Couder Y, Fort E (2006) Single-Particle Diffraction and interference at a macroscopic scale. Phys Rev Lett 154101:97
67. Ibe OC (2014) Fundamentals of applied probability and random processes. 2nd edn. Elsevier, New York
68. Naik SN, Walley SM (2020) The Hall–Petch and inverse Hall–Petch relations and the hardness of nanocrystalline metals. J Mater Sci 55:2661–2681. https://doi.org/10.1007/s10853-019-04160-w
69. Chen R, Matt M, Pauly F, Nielaba P et al (2014) Shot noise variation within ensembles of gold atomic break junctions at room temperature. J Phys Condens Matter 26:474204. https://doi.org/10.1088/0953-8984/26/47/474204
70. Cai Y, Tao Y, Yan Z (2020) Stock market trading volumes and economic uncertainty dependence: before and during Sino-U.S. trade friction. Economic Research-Ekonomska Istraživanja 33(1):1711–1728. https://doi.org/10.1080/1331677X.2020.1758185
71. Calin GA, Vasilescu C, Negrini M, Barbanti-Brodano G (2003) Genetic chaos and antichaos in human cancers. Med Hypotheses 60(2):258–262. https://doi.org/10.1016/s0306-9877(02)00383-3

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Bahdan Zviazhynski is a Physics PhD student at the University of Cambridge, working with Dr Gareth Conduit. Bahdan develops machine learning methods that extract information from noise in physical systems and uses these methods to solve real-world problems in physical sciences. Bahdan graduated from the University of Cambridge, with BA and MSci degrees in Natural Sciences.

Gareth Conduit has a track record of applying artificial intelligence to solve real-world problems. The approach, originally developed for materials design, is now being commercialized by startup Intellegens in not only materials design, but also healthcare and drug discovery. Previously, Gareth had research interests in strongly correlated phenomena, in particular proposing spin spiral state in the itinerant ferromagnet that was later observed in CeFePO. Gareth's group is based at the University of Cambridge.