J Mater Sci

Computation & theory

Exploring supervised machine learning for multi-phase identification and quantification from powder X-ray diffraction spectra

Jaimie Greasley[1,*] and Patrick Hosein[2]

[1] *Department of Physics, The University of the West Indies, St. Augustine, Trinidad*
[2] *Department of Computer Science, The University of the West Indies, St. Augustine, Trinidad*

Received: 17 November 2022
Accepted: 17 February 2023

ABSTRACT

Powder X-ray diffraction analysis is a critical component of materials characterization methodologies. Discerning characteristic Bragg intensity peaks and assigning them to known crystalline phases is the first qualitative step of evaluating diffraction spectra. Subsequent to phase identification, Rietveld refinement may be employed to extract the abundance of quantitative, material-specific parameters hidden within powder data. These characterization procedures are yet time-consuming and inhibit efficiency in materials science workflows. The ever-increasing popularity and propulsion of data science techniques has provided an obvious solution on the course toward materials analysis automation. Deep learning has become a prime focus for predicting crystallographic parameters and features from X-ray spectra. However, the infeasibility of curating large, well-labeled experimental datasets means that one must resort to a large number of theoretic simulations for powder data augmentation to effectively train deep models. Herein, we are interested in conventional supervised learning algorithms in lieu of deep learning for multi-label crystalline phase identification and quantitative phase analysis for a biomedical application. First, models were trained using very limited experimental data. Further, we incorporated simulated XRD data to assess model generalizability as well as the efficacy of simulation-based training for predictive analysis in a real-world X-ray diffraction application.

Handling Editor: Ghanshyam Pilania.

Address correspondence to E-mail: jaimie.greasley@gmail.com
E-mail Address: patrick.hosein@sta.uwi.edu

https://doi.org/10.1007/s10853-023-08343-4

Published online: 15 March 2023

Introduction

The acquisition and assessment of X-ray diffraction (XRD) spectra is the central tenet of many materials characterization investigations. For a material of interest, X-ray scattering data reveals structural and microstructural parameters in addition to essential information on the intrinsic symmetrical arrangement of atoms in lattices [1]. The collection of scattered intensities corresponding to a set of interplanar *d*-spacing magnitudes serves as a fingerprint reference for the material structure [2]. Justifiably, the diffraction spectra of powdered materials are compiled into large databases consisting up to hundreds of thousands of records for experimental reference [3].

The task of matching diffraction peaks to known crystalline phases is not without effort. Regardless the use of search-match software, in the absence of contextual information on the nature of the sample or a great deal of expertise, phase identification is far from automatic [4]. Differing instrument settings and sample-specific features produce altered versions of the expected diffraction profile for a given crystalline phase. This obscures the process of cross-referencing against the database records. Phase matching is further obstructed when dealing with powder patterns of poorly crystallized or multi-phase materials due to exacerbated peak overlap. It is time-inefficient, bordering on impractical, to sieve through the numerous search-match hits for visual assessment of each record, and then to fill in the blanks via trial-and-error to account for unassigned peaks. Moreover, wherever a complete characterization of the sample is needed, it is vital to perform a whole powder pattern fitting (WPPF) routine like the Rietveld method [5]. The latter permits the extraction of refined material parameters including phase fractions, lattice dimensions, atom positions, occupancy factors, crystallite size, texture and strain. Despite the magnitude of data retrieved, Rietveld refinement from powder X-ray data can be tedious, sometimes requiring an hour or more to complete [6, 7].

Attention has already been drawn to such impediments in materials characterization, and with particular concern in the context of high-throughput experimentation (HTE). For instance in thin-films, the high research capacity gained by state-of-the-art methods to rapidly synthesize and screen large combinatorial libraries is bottle-necked at the stage of data analysis [8, 9]. However, the genesis of exploring machine learning (ML) for materials analysis automation began as a result of the inclination to overcome such problems. In the same manner, the launching of the Materials Genome Initiative (MGI) [10] has propelled an ease-of-access to both synthetic and experimental datasets within the last decade. These foundations have culminated with an explosion of data-driven research in the materials science domain with the evolution of materials informatics, also deemed "the fourth paradigm" [11].

Within the framework of supervised machine learning algorithms, exceptional emphasis has been placed so far on deep learning (DL) for predicting various material properties from crystallographic data [12]. In the line of X-ray diffraction profile analysis, work has been done on crystal system prediction [13–16], space group determination [7, 13, 14] as well as indicating the presence of various crystalline phases [17–21]. For phase classification, Lee et al. [17] used a convolutional neural network (CNN) to discern 38 inorganic phases within a quaternary system. The model was trained with about 1.8 million simulations. Similarly, Szymanski et al. [21] executed an ensemble CNN trained with around 38 thousand synthetic profiles for phase identification in monophasic and multi-phasic compositions. The estimation of quantitative structural and compositional parameters is the alternate objective of deep learning research for diffraction data analysis. CNNs have been employed also in the interest of lattice parameter [22, 23], crystallite size [23] and phase fraction prediction [17, 19].

A deep learning approach can afford high accuracy without a requirement of data preprocessing or feature engineering. Yet, DL is also criticized for lack of interpretability due to an inherently complex construction which renders it difficult to extract contextual rule-based knowledge [11]. The greater drawback for deep learners is, however, the considerable number of training parameters which call for labeled 'big' data and a substantial amount of time for any effective training of the model. In many scenarios, well-characterized 'big' XRD data is only conceivable by supplementing experimental spectra with simulated data.

Maffettone et al. [20] described an ensemble system composed of 50 CNNs for phase identification. While the model does not seek explicitly labeled input data

from the user, it requires hours to construct and train the ensemble with synthetically generated spectra from a pre-defined phase library. One study by Lee et al. [19] sought to boost the previous report's CNN accuracy [17] for phase fraction prediction. They investigated for 21 compounds of a quartenary system relevant to solid-state electrolytes. Oddly enough, after augmenting the dataset to nearly 14 million simulations, the authors still reported poor performance for deep network architectures and concluded that a single hidden layer neural network gave higher accuracy. Better results were even seen with conventional learning models like Random Forest (RF), *k*-Nearest Neighbors (k-NN) and the Support Vector Machine (SVM).

On reviewing the existing literature, it has become evident that simpler learning models are often overlooked without reasonable domain-specific or empirical justification. Few studies [24, 25] have endeavored first to investigate elementary supervised learning algorithms like Nearest Neighbors, Support Vector Machines, Decision Trees, Naive Bayes and ensemble techniques for multi-label phase identification of XRD data. Models like these are formulated by hypothesizing some expected property or structure to the data prior to training, whereas neural networks attempt to learn the structure entirely, thereby requiring many more training examples. In the present paper, we have taken an interest in evaluating the application of elementary supervised learning algorithms rather than deep learners to the task of predicting mineral phase composition for a medical application. What is also relevant to our aim is estimating the relative phase weights in multiphasic compositions. We have applied several regression models to this task and evaluated their performance. Below is an outline of the biomedical application context.

Kidney stone analysis

Stone disease is an intensely painful condition with prevalence falling usually in the 5-10% range for a population [26]. Kidney stones or more precisely urinary tract calculi, arise from the nucleation, growth and clustering of mineralogical crystals propelled by persistent supersaturation of constituent ions in urine. Kidney stones often exhibit a high degree of crystallinity [27] and are composed of mainly inorganic phases such as calcium-based oxalates (CaOx) and phosphates (CaPh) as well as magnesium-based phosphates. The less common stone compositions are uric acid, urates and protein phases.

The formation of calculi in the urinary tract is an aberrant event catalyzed by some anatomical, genetic, metabolic, dietary or environmental anomaly for the individual [28]. Resolving the exact composition of a stone often gives strong indicators to the pathology driving its formation. This guides medical practitioners in diagnosing the underlying condition and deriving a target-specific treatment plan. Given a first-time stone event, there is an elevated risk for another stone. The presence of specific mineral phases is also a pointer to the likeliness and degree of recurrent disease. In Table 1, some common kidney stone minerals, their frequencies, associated pathologies and risk for recurrence are outlined.

For the reasons outlined above, stone analysis is highly recommended for all first-time patients by the major international urological associations [29, 30]. Moreover, stone analysis via powder X-ray diffraction is by far the most accurate and sophisticated method for the precise identification of mineral phase composition in kidney stones. Recently, we reported XRD Rietveld characterization of mineral phases in a batch of 46 urinary tract stones [31]. We identified 7 distinct phases throughout the study, with an average of 2.2 phases being detected in each stone. The maximum number identified was 4 phases per sample which was the case for 13% of the batch. Given that quantitative Rietveld analysis becomes increasingly laborious and time-consuming with each additional phase, it is warranted that we have sought an automated approach for a routine stone analysis program. Here, we have furthered our investigation to evaluate the performance of supervised learning models for identifying and quantifying mineral phases from powder X-ray diffraction spectra in context of the described application.

Experimental methods

Data preparation

Urinary tract calculi were previously collected from consenting adult patients at public health institutions. Powder X-ray diffraction spectra were measured with a Bruker D2 Phaser bench-top diffractometer for

Table 1 Frequency, Recurrence Risk and Pathological Associations for Minerals found in Kidney Stones [28, 32, 33]

Mineral phase	Frequency	Risk level	Common associations
Calcium oxalate monohydrate (Whewellite)	78%	Typically low, except for primary hyperoxaluria	Primary hyperoxaluria, secondary hyperoxaluria, Randall's plaque, inflammatory bowel disease, chronic pancreatitis
Calcium oxalate dihydrate (Weddellite)	48%	Low	Hypercalciuria, hypocitraturia, primary hyperthyroidism
Carbonated hydroxyapatite	33%	Low to medium, high for dRTA	Distal renal tubular acidosis (dRTA), hypercalciuria, urinary tract infection (UTI)
Magnesium ammonium phosphate hexahydrate (Struvite)	6%	Medium to high	Urinary tract infection (UTI) by urea splitting organisms
Uric acid (Uricite)	10%	Medium to high	Low urine pH, insulin resistance, type II diabetes, metabolic syndrome, morbid obesity
Ammonium acid urate	1%	Medium to high	Hyperuricusoria, urinary tract infection, chronic diarrhea, laxative abuse
Calcium hydrogen phosphate dihydrate (Brushite)	1-2%	High	Hypercalciuria, primary hyperthyroidism, recurrent stones
Cystine	<1%	High	Cystinuria

an angular range $2^\circ \leq 2\theta \leq 55^\circ$ at step-size of 0.02°. Preliminary phase identification was performed by means of search/match in DIFFRAC.EVA (v4.2) utilizing the International Center for Diffraction Data (ICDD) PDF-2 database. Whole profile fitting was achieved with MAUD [34] Rietveld refinement software for verification of phase presence as well as for obtaining optimized sample parameters including weight fractions.

Synthetic XRD patterns were prepared with a variety of methods. The theoretical line spectra of those phases which we previously found to have larger crystallites were initiated with *pymatgen* middleware [35] using the relevant structural CIF files obtained from the Crystallography Open Database [36]. The physics-based data augmentation protocol by Szymanski et al. [21] was then applied for introducing zero offset, non-uniform lattice parameter and strain-related variance of peak positions, line spectra broadening due to finite crystallite sizes and modified peak intensity signals arising from preferred orientation. The addition of a 4th to 7th order background polynomial with random noise was another experimental effect we incorporated. It was also necessary to replicate the broadened overlapping peaks that were observed experimentally for phases with smaller than usual and even nanometric crystallite dimensions. For these phases, GSAS-II software [37] was utilized for realistic XRD simulation. Multi-phase patterns were fabricated by linearly combining weighted monophasic spectra. The different data collection and preparation methods meant that we were unable to select appropriate scale factors for the simulated data. For consistency, both the experimental and simulated datasets were normalized relative to the maximum signal value of each spectrum. Figure 1 summarizes the data preparation method.

Supervised models

We investigated several supervised learning models and a couple ensemble methods for multi-label phase classification: *k*-Nearest Neighbors (**k-NN**), Support Vector Machine (**SVM**), Decision Tree (**DTr**), Gaussian Naive Bayes (**GNB**), Multinomial Naive Bayes (**MNB**), Complement Naive Bayes (**CNB**), Random Forest (**RF**) and Extremely Randomized Trees (**ExTr**). The optimized hyper-parameters for each model were selected according to the Scikit-learn [38] GridSearchCV method tested on randomized sets of the data. For comparison, we also trained a shallow artificial neural network (**ANN**). The ANN comprised 1000 nodes in its single hidden layer with logistic function activation and used the ADAM optimization algorithm for training. Other network architectures were initially tried on simulated data by varying the number of nodes in the hidden layer by increments of 100. We also tested one double hidden

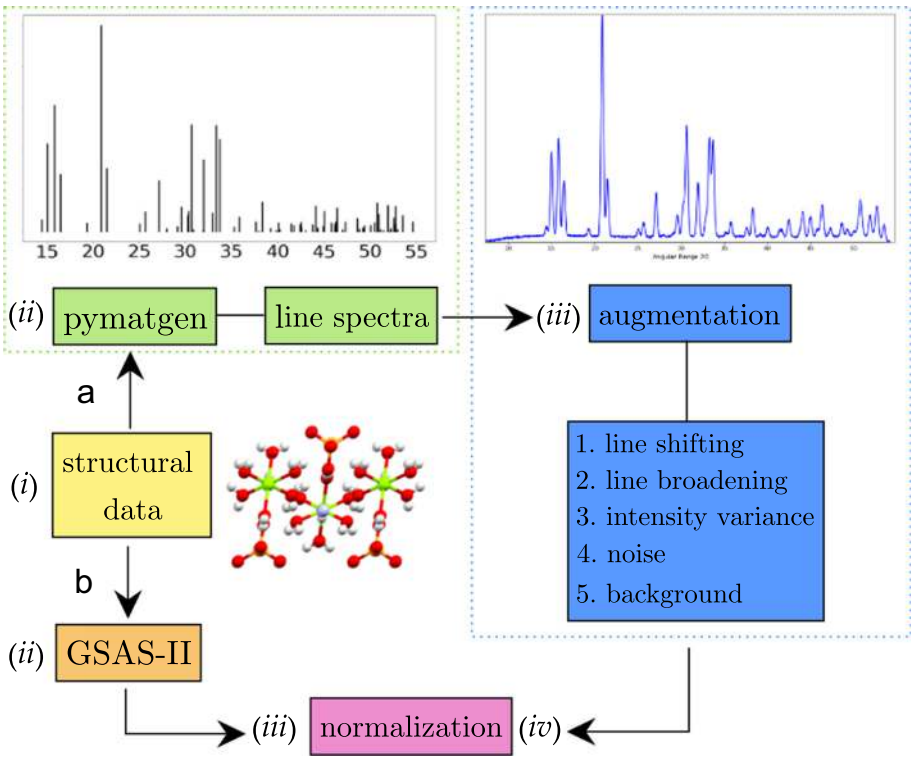

Figure 1 A schematic of the X-ray diffraction spectra simulation process via (a) pymatgen and (b) GSAS-II..

layer network structure. With fivefold cross-validation, the performances of the single HL networks were relatively similar, with the 1000-node ANN giving only a slightly better average overall. We persisted with this construction for the rest of the study. For the second objective of predicting the weight fractions of crystalline phase labels, several of the above classification algorithms like tree-based methods and nearest neighbors already are inherently structured for multi-output regression. We were able to test equivalent multi-target regression versions for the k-NN, SVM, DTr, RF and ExTr algorithms as well as ANN.

Evaluation metrics

Regarding the multi-label classification problem, we utilized the standard metrics like accuracy, precision and recall. Accuracy is defined in the conventional manner as the sum of true positive (TP) and true negative (TN) predictions divided by the total number of predictions. Precision and recall take into account false positive and false negative predictions, respectively. A high false positive (FP) rate is homologous with low precision, whereas a high false negative (FN) rate correlates with low recall. There is typically a trade-off made in assessing model performance with these two metrics as modification of various hyper-parameters to improve one often results in a declined performance for the other. A given application context may emphasize the relative importance of high precision versus that of high recall required of the predictive model. In stone analysis, the inability to recognize certain high-risk phases even in small quantities can lead to an inaccurate diagnosis of the problem and inadequate treatment for recurrence prevention. Yet, for the more common phases that carry lower risk for recurrence and complications, a false negative prediction may not necessarily modify the overall conclusion of the analysis. On the other hand, a model that is not very precise is practically ineffective.

The F_1 score metric places equal importance on precision and recall [eq. (1)]. There are several methods for calculation, namely 'macro', 'micro', 'weighted' and 'samples'. Consider T testing samples and the ascribed K labels. The micro F_1 method relates to the total precision and recall of the $T \times K$ predictions, while macro F_1 considers precision and recall for each of the K labels singly then takes the average. In the case of an imbalanced dataset, the weighted F_1 score is preferred to the macro as it gives more credit to the precision-recall values of the better

represented labels in the dataset. The samples F_1 score is applicable to multi-label problems where each sample can be predicted positive for more than one of the K classes. The precision–recall value for each sample is evaluated, and the average is taken across the T samples.

$$F_1 = 2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

For quantitative phase estimations, three regression metrics were applied for performance evaluation. Considering a single test sample, we denote the vectors of the true and predicted phase fractions as α and $\hat{\alpha}$, respectively. The mean absolute error (MAE) [eq. (2)] enables a direct quantification of the average difference between the real and projected values for each prediction made by the model.

$$\mathbf{MAE} = \frac{1}{\mathrm{TK}} \sum_{s,j} \left| \alpha_{s,j} - \hat{\alpha}_{s,j} \right| \tag{2}$$

The cosine similarity metric (CS) measures the extent to which two vectors point in the same direction, irrespective of their norm magnitudes. The cosine similarity is expressed as in eq. (3).

$$\mathbf{CS} \equiv \cos\theta = \frac{\alpha \cdot \hat{\alpha}}{||\alpha|| \times ||\hat{\alpha}||} \tag{3}$$

For our application, CS may be useful for outlining similarity in the relative ratios between predicted phases rather than their exactly predicted quantities. Two vectors $\vec{a} = [1, 2]$ and $\vec{b} = [4, 8]$ give a perfect cosine similarity of 1 as the relative ratio of the second vector component to the first is 2:1, i.e., their slopes are equal.

We now define the performance metric ρ [eq. (4)] which accounts for both the magnitudes and relative directions of the predicted and true label vectors. The Euclidean distance between the two vectors is represented as $||\alpha - \hat{\alpha}||_2$. As the elements in each vector α and $\hat{\alpha}$ must sum to 1, the maximum value of the Euclidean distance is $\sqrt{2}$, the case of orthogonal vectors.

$$\rho = 1 - \frac{||\alpha - \hat{\alpha}||_2}{\sqrt{2}} \tag{4}$$

If therefore $\alpha = \hat{\alpha}$, we get $\rho = 1$ which implies a perfect prediction. If there is only one phase and the model predicts any other phase except the true phase, then $\rho = 0$ indicating a wrong prediction. Consider now a bi-phasic example where two phases are correctly predicted, but the true fractions are 0.5 and 0.5 and the predicted fractions are 0.6 and 0.4. We obtain $\rho = 0.9$ indicating a close but not perfect prediction. While the ρ metric is ideal for the problem, regression algorithms are not explicitly constrained to maintain positive, normalized outputs, though this may be the expectation given the input training data.

Results and analysis

Phase identification–multi-label classification

First was to evaluate whether traditional learning algorithms are useful for indicating the presence of crystalline phases from experimental powder data. Supervised learning models selected for this part of the study were those already adapted for multi-label function using continuous-variable features in the Scikit-learn library. These were trained until convergence prior to testing. We carried out four instances of assessment: [1] simulation training with simulation testing on the primary and secondary simulated datasets, [2] experimental training with experimental testing, [3] simulation training using the primary simulated dataset with experimental testing and lastly [4] training using the primary simulated dataset and randomly selected experimental examples followed by testing with the remaining experimental data. Our primary simulated dataset consisted of 105 monophasic spectra augmented from the 7 crystalline phases that were identified for the stone batch previously [31]. The secondary dataset (N=184) comprised the original in addition to several weighted linear combinations of the primary spectra that were sensible given the medico-chemical context. The central focus of the study is yet the raw performance of the models without the use of any simulation data, i.e., instance 2. In all instances, training was roughly 80% of the data and 20% left for testing.

In Tables 2 and 3, the summarized results are displayed. Table 2 contains model performances for the first two instances and Table 3 shows the last two. Figure 2 depicts results for instance 2 alone. We present the F_1 micro (mic), weighted (wtd) and samples (smps) scores and their average (avg). F_1 macro scores were omitted as several phase labels are naturally less common than others [Table 1]. Accuracy is registered as three types: the exact-label

Table 2 Model Performance for Simulation-only and Experimental-only Training and Testing

Model*	[1] Simulation-only training & testing								[2] Experimental-only training & testing						
	DT	F_1 Scores				Accuracy			F_1 Scores				Accuracy		
		mic	wtd	smps	**avg**	ELA	MPA	Total	mic	wtd	smps	avg	ELA	MPA	Total
k-NN	s	0.804	0.765	0.695	0.755	0.695	0.695	0.952	0.768 (0.792)	0.739 (0.766)	0.785 (0.798)	0.764	**0.418**	0.927	0.875 (0.894)
	m	0.742 (0.749)	0.721 (0.731)	0.718 (0.729)	0.727	0.416 (0.459)	0.827	0.895 (0.900)							
SVM	s	**0.944**	**0.932**	0.895	**0.924**	**0.895**	0.895	**0.985**	0.781 (0.796)	0.772 (0.800)	0.805 (0.813)	0.786	**0.473 (0.509)**	**0.982**	0.873 (0.886)
	m	**0.828 (0.834)**	**0.824 (0.832)**	**0.809 (0.818)**	**0.820**	**0.562 (0.589)**	0.924	**0.927 (0.931)**							
DTr[a]	s	0.629	0.644	0.629	0.634	0.629	0.629	0.894	0.681	0.680 (0.684)	0.691	0.684	0.291	0.818	0.810 (0.813)
	m	0.579	0.564	0.562	0.568	0.276 (0.292)	0.616	0.822							
DTr[b]	s	0.638	0.639	0.638	0.638	0.638	0.638	0.897	0.716	0.718 (0.732)	0.736	0.723	0.345 (0.364)	0.873	0.836 (0.844)
	m	0.631	0.624	0.616 (0.618)	0.624	0.357 (0.384)	0.703	0.842 (0.845)							
GNB	s	**0.964**	**0.959**	**0.933**	**0.952**	**0.933**	**0.933**	**0.990**	0.625	0.714 (0.734)	0.617	0.652	0.145 (0.182)	0.891	0.730 (0.738)
	m	0.658	0.645	0.653	0.652	0.297	0.849	0.821							
MNB	s	0.923	0.912	0.863	0.899	0.857	0.867	0.98	0.795 (0.798)	**0.808 (0.828)**	0.796	0.8	0.382	0.945	**0.883 (0.891)**
	m	0.818	0.817 (0.819)	0.805 (0.806)	0.813	0.503 (0.514)	**0.941**	0.917							
CNB	s	0.910	0.915	**0.902**	0.909	0.876	**0.914**	0.974	**0.808 (0.811)**	**0.820 (0.842)**	**0.807**	**0.812**	0.382	**0.964**	**0.888 (0.896)**
	m	0.810	0.815	0.803	0.809	0.443	**0.968**	0.908							
RF[c]	s	0.706	0.632	0.552	0.63	0.552	0.552	0.935	0.769 (0.782)	0.746 (0.770)	0.783 (0.786)	0.766	0.364	0.927	0.873 (0.886)
	m	0.641 (0.642)	0.590 (0.591)	0.541	0.591	0.330 (0.362)	0.578	0.879 (0.883)							
ExTr[c]	s	0.759	0.723	0.619	0.700	0.619	0.619	0.944	0.774 (0.788)	0.760 (0.787)	0.780 (0.784)	0.771	0.345 (0.364)	**0.964**	0.870 (0.883)
	m	0.649 (0.654)	0.576 (0.580)	0.570 (0.572)	0.598	0.330 (0.362)	0.589	0.879 (0.884)							
ANN	s	0.938	0.912	0.886	0.912	0.886	0.886	0.984	**0.802 (0.808)**	0.807 (0.822)	**0.820**	**0.810**	**0.473 (0.509)**	**0.982**	**0.883 (0.891)**
	m	**0.837 (0.839)**	**0.826 (0.831)**	**0.817 (0.824)**	**0.827**	**0.568 (0.595)**	0.924	**0.932 (0.934)**							

[a]Using Gini optimization

[b]Using entropy optimization

[c]For [1] Gini optimized with 70 estimators; for [2] entropy optimized with 70 estimators

The left side shows data for instance 1 where models are trained and tested on simulations only and the right side for instance 2 with real, experimental data only. F_1 and accuracy scores are the average of fivefold cross-validation. 'ELA' stands for exact-label accuracy. 'MPA' is the majority phase accuracy. 'DT' is the dataset type for simulations only, i.e., only single phase denoted by 's' for which there were 105 spectra, or both single and multi-phase denoted by 'm' which consisted of 184 spectra. Values in parentheses ( ) show any improved performance if phases in quantities less than 10% were ignored. Scores in **bold** highlight the two highest values for each metric and category. ***k-NN** *k*-Nearest Neighbors, **SVM** Support Vector Machine, **DTr** Decision Tree, **GNB** Gaussian Naive Bayes, **MNB** Multinomial Naive Bayes, **CNB** Complement Naive Bayes, **RF** Randomized Forest, **ExTr** Extremely Randomized Trees, **ANN** Artificial Neural Network (Multi-Layer Perceptron)

Table 3 Model Performance for Testing on Experimental Data using an Augmented Training Dataset

Model*	[3] Simulation-only Training for Experimental Testing							[4] Simulation + Experimental Training for Experimental Testing						
	F_1 Scores				Accuracy			F_1 Scores				Accuracy		
	mic	wtd	smps	avg	ELA	MPA	Total	mic	wtd	smps	avg	ELA	MPA	Total
k-NN	0.72 (0.74)	0.68 (0.70)	0.76 (0.77)	0.72	0.346	0.923	0.857 (0.874)	0.712 (0.734)	0.680 (0.700)	0.730 (0.742)	0.707	0.348	0.865	0.855 (0.871)
SVM	**0.79**	**0.78 (0.79)**	**0.79**	**0.787**	0.346	**1.000**	**0.879 (0.885)**	**0.806 (0.820)**	**0.791 (0.807)**	**0.822 (0.831)**	**0.806**	**0.484 (0.506)**	**0.984**	**0.888 (0.901)**
DTr[a]	0.60	0.58	0.61	0.597	0.192	0.712	0.755	0.646 (0.648)	0.631 (0.637)	0.655 (0.659)	0.644	0.297 (0.313)	0.758	0.793 (0.804)
GNB	0.71	0.71 (0.72)	0.71	0.710	0.365 (0.385)	0.865	0.824 (0.835)	0.601	0.632	0.625	0.619	0.213 (0.229)	0.861	0.726
MNB	**0.78 (0.79)**	**0.77 (0.79)**	**0.78**	0.777	**0.442 (0.481)**	0.942	0.874 (0.885)	0.702 (0.741)	0.679 (0.722)	0.728 (0.756)	0.703	0.397 (0.413)	0.916	0.849 (0.876)
CNB	**0.79 (0.80)**	**0.78 (0.80)**	**0.78 (0.79)**	**0.783**	**0.423 (0.481)**	**0.962**	**0.876 (0.887)**	0.727 (0.758)	0.708 (0.744)	0.746 (0.769)	0.727	0.406 (0.435)	0.929	0.856 (0.879)
RF[b]	0.54 (0.55)	0.51 (0.52)	0.53	0.527	0.212	0.635	0.786 (0.802)	0.642 (0.671)	0.588 (0.618)	0.643 (0.660)	0.624	0.326 (0.339)	0.774	0.827 (0.851)
ExTr[c]	0.55	0.50	0.54	0.530	0.250	0.577	0.797 (0.808)	0.673 (0.690)	0.636 (0.657)	0.666 (0.679)	0.658	0.329	0.800	0.829 (0.847)
ANN	0.76 (0.79)	0.75 (0.78)	**0.78 (0.80)**	0.763	0.404 (0.423)	**1.000**	0.871 (0.893)	**0.809 (0.819)**	**0.794 (0.806)**	**0.825 (0.832)**	**0.809**	**0.494 (0.523)**	**0.984**	**0.889 (0.901)**

[a]Entropy optimization

[b]Entropy optimization with 70 estimators

[c]Gini optimization with 70 estimators

The left side of the table reflects results from a single iteration of training with the secondary simulated dataset (N=184) comprising monophase and multi-phase synthetic spectra. The test samples were the complete experimental dataset (T=52). The right side of the table shows averaged results from 10 iterations of training with 105 simulated single-phase spectra and 21 randomly selected experimental scans. Testing was performed on the remainder T=31 experimental spectra. Values in parentheses ( ) show any improved performance if phases in quantities less than 10% were ignored. Scores in **bold** highlight the two highest values for each metric and category.

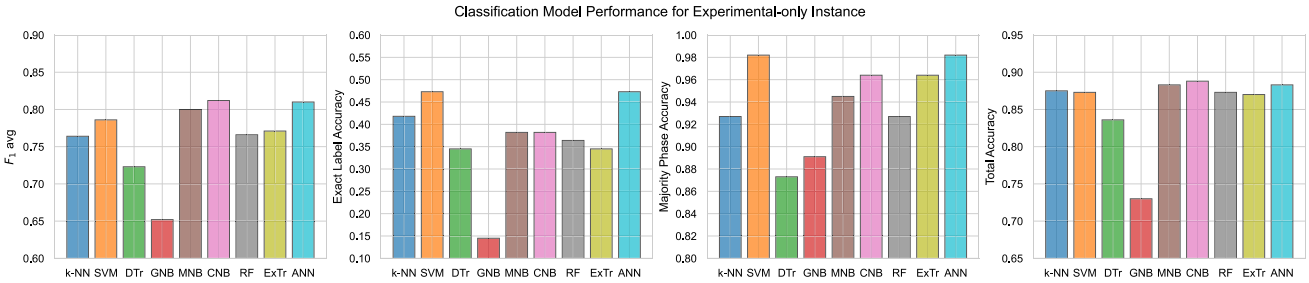

Figure 2 Experimental-only Classification Performance.

accuracy (ELA), the majority phase accuracy (MPA) and the total accuracy. The total accuracy is the standard definition which considers the overall correctness of all the predictions made by the classifier. The MPA is the percentage of samples whereby at least the majority phase composition is predicted positively. The ELA is the percentage of samples in the dataset where all phases are exactly predicted. That is to say, if 3 of 7 possible phases exist in the sample, then these are predicted as '1', while all others predicted as '0'. In the case where one of the phases is not predicted or where an additional phase is predicted outside of the original 3, this is rendered as an incorrect prediction. As to be expected, ELA values were lower than the MPA and total accuracies as the qualification criterion is quite stringent. In the case where the entire experimental set was reserved for testing, i.e., instance 3, the scores reflected are for a single evaluation. Otherwise, for instances 1 and 2, F_1 and accuracy scores were the average of fivefold cross-validation and for instance 4 the average of 10 evaluations.

The simulation-only evaluation, i.e., instance 1, was split for the primary and secondary simulation datasets. The Gaussian Naive Bayes (GNB) classifier performed exceptionally with a total accuracy of 99% (F_1 avg = 0.952, MPA = 0.933) for monophasic spectra which implies a Gaussian spread of the characteristic features in each phase for simulated data. The Support Vector classifier (SVM) as well as the other Naive Bayes models also showed a strong performance alongside the shallow artificial neural network (ANN). The SVM attained the second highest F_1 average and total accuracy with 0.924 and 98.5% respectively. The Complement Naive Bayes (CNB) method had the second best majority phase accuracy MPA = 0.914. Although classifying single-phase spectra may not appear a major challenge, the simulation protocol produced significant alterations to peak intensity, position and widths for each class. The ability to still recognize a phase despite these large changes is noteworthy. Appending the multi-phasic simulations, we see a drastic decrease for the GNB classifier but maintained performance of ANN, SVM and other Naive Bayes classifiers. In this case, the ANN scored highest with F_1 avg = 0.827 and a 93.2% total accuracy, but SVM was a very close second with F_1 avg = 0.820 and 92.7% total accuracy. The exact label was also predicted quite well for both models with 56.8% and 56.2% for ANN and SVM accordingly. It may be said that SVM performed on par with the ANN despite requiring a substantially shorter training time, which is 0.08 seconds compared to 88 seconds for the ANN with this dataset. With regards to accurately identifying the majority phase, the other two Naive Bayes classifiers topped the SVM and ANN with MPA = 0.968 for CNB and MPA = 0.941 for Multinomial Naive Bayes (MNB).

For instance 2 which dealt solely with experimental spectra, all classifiers showed good ability to positively predict the majority phase. To be sure, the experimental dataset exhibited less varied features in each class label than the simulated data. Hence, models which performed relatively poorly on the synthetic spectra performed better on actual experimental data. For example, the tree-based classifiers gave the lowest majority phase accuracies in instance 1 between 57.8% for Random Forest (RF) and 70.3% for the entropy-optimized Decision Tree (DTr). These values rose to 87.3% and 92.7% in instance 2. The best MPA for instance 2 was 98.2% attained by both SVM and ANN. The exact-label accuracy was also the highest for these two models with all phases being correctly assigned for 47.3% of samples. In terms of overall performance, the CNB classifier attained the

best F_1 average and total accuracy of 0.812 and 88.8% accordingly. This was followed very closely by the ANN with F_1 avg = 0.810 and 88.3% total accuracy. It must again be mentioned that the training time for CNB was a much shorter 0.03 seconds to 67 seconds for the ANN.

Table 3 records the data for instances 3 and 4 which tests exclusively on experimental data but integrates the synthetic profiles for training. For instance 3, the majority phase of all testing samples were perfectly predicted by the SVM and ANN. The next best was 96.2% for the CNB classifier. SVM outperformed ANN with F_1 avg = 0.787 versus 0.763, and only by a little in total accuracy with 87.9% versus 87.1%. The Complement and Multinomial Naive Bayes' methods both also did better than the ANN in terms of F_1 average. The exact-label accuracy was highest for MNB and CNB with 44.2% and 42.3%, respectively. Lastly, a fraction of the experimental data was subsumed in the primary simulation training set for instance 4. We observe the highest scoring by the ANN and SVM classifiers once more at F_1 avg = 0.809 with 88.9% total accuracy against F_1 avg = 0.806 and 88.8% accordingly. Second to these was CNB, though there was a solid margin between Naive Bayes and the former two.

Results recorded for instances 2 to 4 are most relevant to practice. From these, traditional models gave the best results collectively in instance 2 for MPA and F_1 score regardless of limited experimental training data. Expectantly, when synthetic data was supplemented for experimental data in training the collective performance of the classifiers dropped. With both SVM and ANN, we note that the prediction accuracy of the majority phase solidified more with added synthetic training. This was not true for other models. We particularly observe that overall ANN and SVM gave the best result in instance 4. The case with the ANN was that though MPA increased from instance 2 to 3, F_1 decreased implying that synthetic-only training made discerning secondary and minor phases more difficult.

In summary, tree-type classifiers consistently gave lower performances such that the Random Forest and Extremely Randomized Trees (ExTr) ensemble methods rarely provided any benefit over a single Decision Tree estimator. Only in instance 2 an advantageous margin developed for the ensembles in MPA, but still with median F_1 scores. The Gaussian Naive Bayes classifier typically showed the poorest performance when tested on real, multi-phasic spectra. Nearest Neighbors (k-NN) had average performance with experimental-only data and ranked 4th in traditional classifiers under the MNB, CNB and SVM in instance 3 and 4. The Support Vector Machine was the best overall traditional model which performed comparably with or even slightly superior to the shallow Artificial Neural Network, yet training in only hundredths of a second. Complement Naive Bayes was second to SVM. As a final note, the CNB and MNB models did perform better with experimental-only training, but the SVM took over once the training set was increased.

Estimation of phase weight fractions–multi-output regression

We replicated the same training-testing scenarios of the previous section for the prediction of phase fractions. Given a more restricted experimental dataset (N=46), we instead applied leave-one-out cross-validation (LOOCV) only for instance 2 where one sample at a time was reserved for testing while all others used in training. The results for mean absolute error **MAE**, cosine similarity **CS** and ρ in each instance are recorded in Table 4. Instance 2 results alone are portrayed in Figure 3. Upon inspection of the outputs, the ANN was one of two models that frequently returned negative weight predictions. The normalization requirement for the total phase fractions to sum to 1 was also not generally maintained. The SVM regressor had similar fault. Though not explicitly constrained, the tree-type and k-NN algorithms were able to reflect the positivity and normalized conditions for the output variables based on the training data seen. We enforced post-prediction constraints for ANN and SVR. Tabulated values reflect the performance metrics after these considerations.

The shallow ANN and k-NN regressors stood out across all instances. The k-NN showed significantly better performance than all other models in instances 1 and 3 with $\rho = 0.776$, MAE = 0.068 and $\rho = 0.846$, MAE = 0.048 accordingly. The ANN performed a little better ($\rho = 0.858$, MAE = 0.049) than SVM ($\rho = 0.837$, MAE = 0.052) in experimental-only instance 2. For instance 3, the scores were split as the k-NN model achieved the better mean absolute error (MAE= 0.083) where the ANN had a better ρ of 0.766.

Table 4 Regression Model Performance for Phase Fraction Estimation on Simulated and Experimental Datasets.

Model†	[1] Sim-only			[2] Exp-only			[3] Sim training			[4] Sim+Exp training		
	ρ	MAE	CS	ρ	MAE	CS	ρ	MAE	CS	ρ	MAE	CS
k-NN	**0.7756**	**0.0682**	0.8718	0.8369	0.0520	0.9470	0.7535	**0.0827**	0.8565	**0.8462**	**0.0479**	**0.9539**
SVR *	0.6830	0.1195	**0.8844**	0.7971	0.0773	0.9364	0.7259	0.1036	0.8612	0.7907	0.0804	0.9508
DTr	0.5418	0.1377	0.5774	0.7438	0.0832	0.8211	0.6244	0.1245	0.6362	0.7536	0.0781	0.8349
RF	0.6393	0.1300	0.7863	0.7908	0.0778	0.9029	0.6855	0.1221	0.8122	0.8024	0.0737	0.9157
ExTr	0.6953	0.1079	0.8313	0.8217	0.0635	0.9232	0.7156	0.1085	0.8044	0.8362	0.0601	0.9392
ANN	0.7108	0.1029	0.8738	**0.8582**	**0.0491**	**0.9662**	**0.7660**	0.0879	**0.8845**	0.8036	0.0702	0.9358

Instance [1] reports on fivefold cross-validation on the simulated dataset (N=184) and instance [2] shows the average of leave-one-out cross-validation on the experimental dataset (N=46). Instance [3] relates to testing on the experimental dataset while training only on the simulated dataset. Instance [4] uses all simulated data plus randomly selects half of the experimental data for training (N=207), then tests on the next half of the experimental (T=23). The averages of 10 evaluations corresponding to 10 randomly selected sets from the experimental data are registered. The highest score for each metric and instance is **bolded**. †**k-NN** Nearest Neighbors regressor (k=2, distance weighted); **DTr** Decision Tree regressor (default); **RF** Random Forest regressor (default); **ExTr** Extremely Randomized Trees (estimators=70); **ANN** Multi-Layer Perceptron regressor. ***SVR** Support Vector regressor with radial basis function kernel for instances [1] and [3-4] and with linear kernel (c=1) for instance [2].

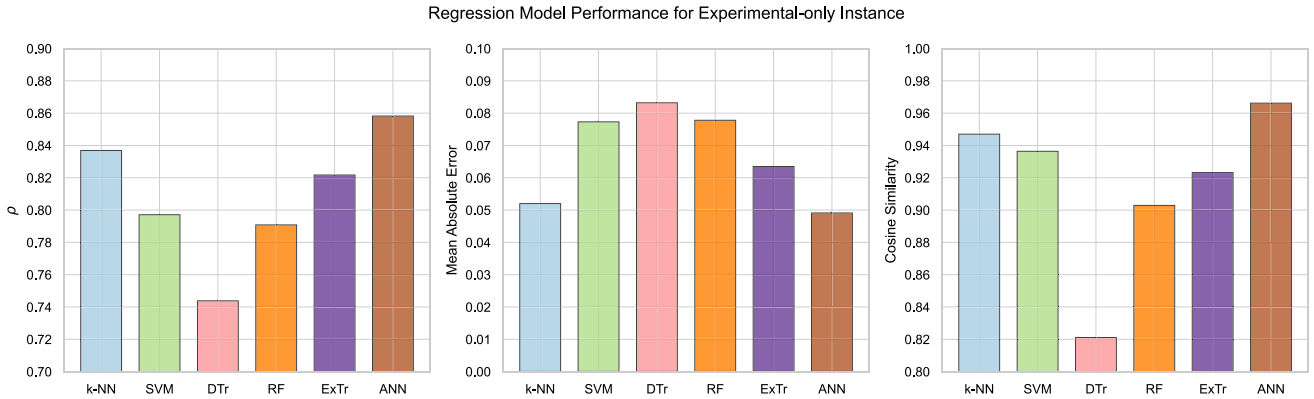

Figure 3 Experimental-only Regressor Performance.

Granted, constraining outputs caused notable improvements for both the ANN and SVR models. Particularly with instance 3, the ANN mean absolute error decreased from 0.118 to 0.088 and cosine similarity increased from CS = 0.876 to 0.885. With the Support Vector, MAE = 0.180 decreased to 0.132 and CS = 0.666 increased to 0.752.

Collectively, regression models performed best to worst in instance 2, 4, 3 then 1 with ρ avg = 0.808, 0.805, 0.712 and 0.674, respectively. Despite only about three dozen training examples, instance 2 gave also the lowest averaged MAE of 0.067. Training only with simulated data, i.e., instance 3, was not ideal even if all major experimental and sample effects were modeled. Inclusion of some experimental data in training saw some advantage over instance 2 for ensemble methods and k-NN alone.

Discussion

Supervised learning investigations for many computational materials analysis tasks including phase identification, predominantly involve the application of deep neural networks. Neural models try to directly *learn* the input-output mapping for a dataset whereas traditional models optimize on the basis of some assumption, be it parametric, methodic or both. Parametric learning models assume some structure or property of the data before parameter optimization through training. For illustration, Linear Regression

models presuppose a linear function between features and their output and then optimize on the gradient weights for this linear structure. With Naive Bayes classifiers, a distribution model is selected, e.g., Gaussian, for evaluating the probabilities of each class given the input data features. For other traditional models, the assumption is less about data structure and more on the approach to group or classify the data. Example, the Support Vector Machine aims to resolve some optimal hyperplane for separating the different class outputs. Any testing data is afterward compared to a set of reference vectors belonging to the different classes which define the boundaries of the plane. SVM may, however, be viewed as parametric with the use of linear or nonlinear kernel functions to derive the hyperplane. The *k*-Nearest Neighbors algorithm takes a "learn by example" approach where the method is simply to label the test point according to the assigned labels of the *k* closest points used in training the model.

Traditional learning algorithms are much faster to train and interpret as the computational approach taken to deduce a representation of the input–output relationship is much more straightforward. They are also extendable to very many applications, failing to perform only if the embedded assumptions do not adequately portray the real problem. In these events, opting for more complex learning models is quite reasonable but presents a different form of challenge. Choosing the neural network architecture is the first as it is well known that there are no standardized rules for it. The experimenter must establish the structural hyper-parameters relating to the type of network, the depth of layers and number of nodes in each layer, before settling on learning hyper-parameters like activation functions, gradient optimization and learning rates. Finding the optimal network structure by training and evaluating outputs with multiple hyper-parameters is a task in itself.

The next difficulty is that training complex architectures with a considerable number of parameters requires 'big' datasets to gain the performance advantage over traditional learners that is anticipated. Yet, it takes time to experimentally acquire materials data and analyzes it for the assignment of labels. Large-scale experimental libraries of well-characterized materials data within a given application are typically unachievable. Consequently, materials research involving deep learning heavily depends on simulation-based training, for which real-world applicability is yet to be ascertained.

In our study, XRD data augmentation served to gauge its applicability in addition to the generalizability of learning models by imposing more variant data features for each phase. Simulated powder spectra can be made to resemble closely to actual scans by modeling various specimen and instrumental effects. For example, finite resolutions of real-world measurement devices ensure that diffraction peaks have determinable widths rather than the infinitesimally narrow theoretical line spectra. Furthermore, peak positions may be varied non-uniformly by altering material parameters for lattice dimensions and macro-strain, peak intensities by including preferred orientation, and peak shapes by setting finite crystallite sizes for each phase. The height of the powdered material on the sample plate can also give rise to a uniform positive or negative angular shift in peak positions for the entire scan. The above described effects as well as the addition of random noise and realistic background functions were all included in data augmentation. It is worth pointing out that the level of effort expended in preparing simulated spectra is dependent on the aim of the investigation, but mostly is at the discretion of the experimenter. For most effective training, we imagine that utter consideration should be made in formulating the parameter bounds for synthetic spectra generation within the applied context, as opposed to overshooting the objective entirely. Still in the present work the best collective classifier performances were achieved for instance 2 which reinforces that the use of experimental data for training is the ideal scenario, even with a small batch.

To sum up, traditional supervised learning algorithms are still tenable for automating tasks in analysis of materials data and ought to be considered first in similar lines of investigation. While some models were not so effective for our biomedical application, others like the Support Vector, Naive Bayes and *k*-Nearest Neighbors contended with the shallow Artificial Neural Network. Other studies regarding XRD phase fraction estimation have likewise reported the success of traditional models over neural networks, even deep learners trained with millions of synthetic spectra [19, 25]. In the current study, we have still not explored dimensionality reduction, nor have we properly ventured into ensemble learning techniques like stacking and boosting for enhancing

model performance. Suzuki et al. [39] reduced simulated XRD line spectra to just 11 features and achieved high accuracy in crystal system prediction with a tree-ensemble classifier. Bunn et al. [24] developed a supervised learning model using AdaBoost [40] for feature extraction from materials spectral data and subsequent phase identification. Further study along these lines may prove beneficial in conclusively demonstrating the full adequacy of traditional models for XRD phase identification and phase fraction estimation tasks.

Conclusion

In the current investigation relating to biomedical materials analysis, we found that the Support Vector Machine (SVM) and Complement Naive Bayes (CNB) classifiers were successful in multi-phase identification with experimental and simulated XRD spectra. For four training-testing scenarios, SVM and CNB were comparable to (and in some instances better than) the Artificial Neural Network despite training in mere hundredths of a second where the ANN took well over a minute. Quantifying relative phase fractions was also successfully executed by a *k*-Nearest Neighbors regressor which performed distinctly better than the ANN regressor in some scenarios and in others only slightly inferior to it. We thus conclude that traditional machine learning models are yet viable choices for task automation in materials analysis applications. They also provide the further advantages of being more interpretable, much faster to train, not requiring enormous training datasets, and having a searchable hyper-parameter space for model tuning.

Data and code availability

Data is available upon request.

Declarations

Conflict of interest The authors have no conflicts of interest to disclose.

Ethical approval The investigation of kidney stones was approved by the University of the West Indies St. Augustine Campus Research Ethics Committee.

References

[1] Dinnebier RE, Billinge SJ (2008) Powder diffraction: theory and practice. R Soc Chem 25:87

[2] Pecharsky V, Zavalij P (2009) Fundamentals of powder diffraction and structure characterization of materials

[3] Gates-Rector S, Blanton T (2019) The powder diffraction file: a quality materials characterization database. Powder Diffr 34(4):352–360

[4] Lutterotti L, Pilliere H, Fontugne C, Boullay P, Chateigner D (2019) Full-profile search-match by the rietveld method. J Appli Crystallogr 52(3):587–598

[5] Rietveld HM (1967) Line profiles of neutron powder-diffraction peaks for structure refinement. Acta Crystallogr 22(1):151–152

[6] McCusker L, Von Dreele R, Cox D, Louër D, Scardi P (1999) Rietveld refinement guidelines. J Appl Crystallogr 32(1):36–50

[7] Oviedo F, Ren Z, Sun S, Settens C, Liu Z, Hartono NTP, Ramasamy S, DeCost BL, Tian SI, Romano G et al (2019) Fast and interpretable classification of small x-ray diffraction datasets using data augmentation and deep neural networks. Comput Mater 5(1):1–9

[8] Mao SS, Burrows PE (2015) Combinatorial screening of thin film materials: An overview. J Materiomics 1(2):85–91

[9] Sun S, Hartono NT, Ren ZD, Oviedo F, Buscemi AM, Layurova M, Chen DX, Ogunfunmi T, Thapa J, Ramasamy S et al (2019) Accelerated development of perovskite-inspired materials via high-throughput synthesis and machine-learning diagnosis. Joule 3(6):1437–1451

[10] de Pablo JJ, Jackson NE, Webb MA, Chen L-Q, Moore JE, Morgan D, Jacobs R, Pollock T, Schlom DG, Toberer ES et al (2019) New frontiers for the materials genome initiative. Comput Mater 5(1):1–23

[11] Agrawal A, Choudhary A (2019) Deep materials informatics: applications of deep learning in materials science. Mrs Commun 9(3):779–792

[12] Choudhary K, DeCost B, Chen C, Jain A, Tavazza F, Cohn R, Park CW, Choudhary A, Agrawal A, Billinge SJ et al (2022) Recent advances and applications of deep learning methods in materials science. Comput Mater 8(1):1–26

[13] Park WB, Chung J, Jung J, Sohn K, Singh SP, Pyo M, Shin N, Sohn K-S (2017) Classification of crystal structure using a convolutional neural network. IUCrJ 4(4):486–494

[14] Vecsei PM, Choo K, Chang J, Neupert T (2019) Neural network based classification of crystal symmetries from x-ray diffraction patterns. Phys Rev B 99(24):245120

[15] Aguiar JA, Gong ML, Tasdizen T (2020) Crystallographic prediction from diffraction and chemistry data for higher

throughput classification using machine learning. Computational Materials Science 173:109409
[16] Zaloga AN, Stanovov VV, Bezrukova OE, Dubinin PS, Yakimov IS (2020) Crystal symmetry classification from powder x-ray diffraction patterns using a convolutional neural network. Mater Today Commun 25:101662
[17] Lee J-W, Park WB, Lee JH, Singh SP, Sohn K-S (2020) A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic xrd powder patterns. Nat Commun 11(1):1–11
[18] Wang H, Xie Y, Li D, Deng H, Zhao Y, Xin M, Lin J (2020) Rapid identification of x-ray diffraction patterns based on very limited data by interpretable convolutional neural networks. J Chem Inf Model 60(4):2004–2011
[19] Lee J-W, Park WB, Kim M, Singh SP, Pyo M, Sohn K-S (2021) A data-driven xrd analysis protocol for phase identification and phase-fraction prediction of multiphase inorganic compounds. Inorganic Chem Front 8(10):2492–2504
[20] Maffettone PM, Banko L, Cui P, Lysogorskiy Y, Little MA, Olds D, Ludwig A, Cooper AI (2021) Crystallography companion agent for high-throughput materials discovery. Nat Comput Sci 1(4):290–297
[21] Szymanski NJ, Bartel CJ, Zeng Y, Tu Q, Ceder G (2021) Probabilistic deep learning approach to automate the interpretation of multi-phase diffraction spectra. Chem Mater 33(11):4204–4215
[22] Chitturi SR, Ratner D, Walroth RC, Thampy V, Reed EJ, Dunne M, Tassone CJ, Stone KH (2021) Automated prediction of lattice parameters from x-ray powder diffraction patterns. J Appl Crystallogr 54:6
[23] Dong H, Butler KT, Matras D, Price SW, Odarchenko Y, Khatry R, Thompson A, Middelkoop V, Jacques SD, Beale AM et al (2021) A deep convolutional neural network for real-time full profile analysis of big powder diffraction data. Comput Mater 7(1):1–9
[24] Bunn JK, Han S, Zhang Y, Tong Y, Hu J, Hattrick-Simpers JR (2015) Generalized machine learning technique for automatic phase attribution in time variant high-throughput experimental studies. J Mater Res 30(7):879–889
[25] Park SY, Son B-K, Choi J, Jin H, Lee K (2022) Application of machine learning to quantification of mineral composition on gas hydrate-bearing sediments, ulleung basin, korea. J Petroleum Sci Eng 209:109840
[26] Qian X, Wan J, Xu J, Liu C, Zhong M, Zhang J, Zhang Y, Wang S (2022) Epidemiological trends of urolithiasis at the global, regional, and national levels: a population-based study. Int J Clin Pract 2022:54
[27] Mirković M, Dosen A, Erić S, Vulić P, Matović B, Rosić A (2020) Phase and microstructural study of urinary stones. Microchem J 152:104429
[28] Daudon M, Dessombz A, Frochot V, Letavernier E, Haymann J-P, Jungers P, Bazin D (2016) Comprehensive morpho-constitutional analysis of urinary stones improves etiological diagnosis and therapeutic strategy of nephrolithiasis. Comptes Rendus Chimie 19(11–12):1470–1491
[29] Pearle MS, Goldfarb DS, Assimos DG, Curhan G, Denu-Ciocca CJ, Matlaga BR, Monga M, Penniston KL, Preminger GM, Turk TM et al (2014) Medical management of kidney stones: Aua guideline. J Urol 192(2):316–324
[30] Turk C, Neisius A, Petřík A, Seitz C, Thomas K, Skolarikos A, (2020) European Association of Urology Guidelines. 2020 Edition., vol. presented at the EAU Annual Congress Amsterdam 2020, european association of urology guidelines. 2020 edition. EAU Guidelines on Urolithiasis 2020. The European Association of Urology Guidelines Office,
[31] Greasley J, Goolcharan S, Andrews R (2022) Quantitative phase analysis and microstructural characterization of urinary tract calculi with x-ray diffraction rietveld analysis on a caribbean island. J Appl Crystallogr 55:1
[32] Schubert G (2006) Stone analysis. Urol Res 34(2):146–150
[33] Daudon M, Jungers P, Bazin D, Williams JC (2018) Recurrence rates of urinary calculi according to stone composition and morphology. Urolithiasis 46(5):459–470
[34] Lutterotti L, Matthies S, Wenk H (1999) Maud: a friendly java program for material analysis using diffraction, IUCr: Newsletter of the CPD, 21(14-15),
[35] Ong SP, Richards WD, Jain A, Hautier G, Kocher M, Cholia S, Gunter D, Chevrier VL, Persson KA, Ceder G (2013) Python materials genomics (pymatgen): a robust, open-source python library for materials analysis. Comput Mater Sci 68:314–319
[36] Gražulis S, Chateigner D, Downs RT, Yokochi A, Quirós M, Lutterotti L, Manakova E, Butkus J, Moeck P, Le Bail A (2009) Crystallography open database-an open-access collection of crystal structures. J Appl Crystallogr 42(4):726–729
[37] Toby BH, Von Dreele RB (2013) Gsas-ii: the genesis of a modern open-source all purpose crystallography software package. J Appl Crystallogr 46(2):544–549
[38] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
[39] Suzuki Y, Hino H, Hawai T, Saito K, Kotsugi M, Ono K (2020) Symmetry prediction and knowledge discovery from x-ray diffraction patterns using an interpretable machine learning approach. Sci Rep 10(1):1–11

[40] Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Computer Syst Sci 55(1):119–139