Spectral Sensor Fusion for Prediction of Li and Zr in Rocks: Neural Network and PLS Methods

Matthew Dirks[a,*], David Turner[b], David Poole[a]

[a] *Department of Computer Science, University of British Columbia, 2366 Main Mall, Vancouver, British Columbia, Canada*
[b] *Turner Geoscience Consulting, British Columbia, Canada*

Abstract

Integrating information from multiple sensors, known as sensor fusion, is particularly challenging for small datasets where selecting from the plethora of available methods poses a significant challenge in chemometric analysis. This study compares several sensor fusion methods (spanning data-level, feature-level, and decision-level fusion) based on partial least squares (PLS) and convolutional neural network (CNN) models. This study is the first to compare simple sensor fusion methods to the latest multiblock PLS models and parallel-input CNNs, to the best of our knowledge. We demonstrate sensor fusion using a small dataset of 177 rock samples on two prediction tasks: predicting lithium (Li) concentrations and zirconium (Zr) concentrations using three types of spectra, namely X-ray fluorescence (XRF), visible to short-wave infrared (Vis-NIR-SWIR), and laser-induced breakdown spectroscopy (LIBS). The best-performing Li model was PLS using XRF (mean RMSEP: 0.078%) and the best sensor fusion model was ROSA (mean RMSEP: 0.087%). The best Zr model was a high-level fusion of PLS models (mean RMSEP: 0.042%).

1. Introduction

Integrating information from multiple, complementary, spectroscopic techniques is particularly challenging for small datasets where ground truth is limited, as is common in spectroscopy. The challenge in chemometric analysis lies in picking an appropriate model. There are many methods for sensor fusion ranging from straightforward models that throw the data together to complex neural network architectures designed specifically for sensor fusion. The aim of this paper is to investigate which methods work well for small data scenarios.

Examples of spectral sensor fusion include combining hyperspectral spectrometers [1], Raman and LIBS [2], and others [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Sensor fusion methods, and multiblock methods [13] more generally, are commonly classified into three levels [14, 10], as shown in Figure 1. Low-level fusion refers to joining data from multiple sources together and analyzing them in the same way as for a single multi-variable source. In mid-level fusion, features are first extracted from each input and then these features are combined. High-level fusion methods combine multiple predictions derived from each input.

In this study, we compare single-sensor methods and methods from each level of sensor fusion on a small dataset. In low-level fusion, the simplest option is to concatenate (after any normalization) all the data together then train a regression model as normal. High-level sensor fusion involves training multiple prediction models and combining their outputs. We consider three high-level sensor fusion methods: a linear model that combines the outputs of (1) partial least squares (PLS) models and (2) neural network models, and (3) a recent multiblock extension to PLS. As for mid-level fusion, neural networks can be arranged to first extract intermediate features from each sensor, then to combine those features together to output a prediction [15, 16]; this model is trained end-to-end [17], which avoids loss of information that can arise when doing vari-

*https://skylogic.ca

Email addresses: mcdirks@cs.ubc.ca (Matthew Dirks), turnergeoscience@gmail.com (David Turner), poole@cs.ubc.ca (David Poole)

Accepted Manuscript; the Published Journal Article is available at https://doi.org/10.1016/j.chemolab.2023.104915 in the journal of Chemometrics and Intelligent Laboratory Systems.

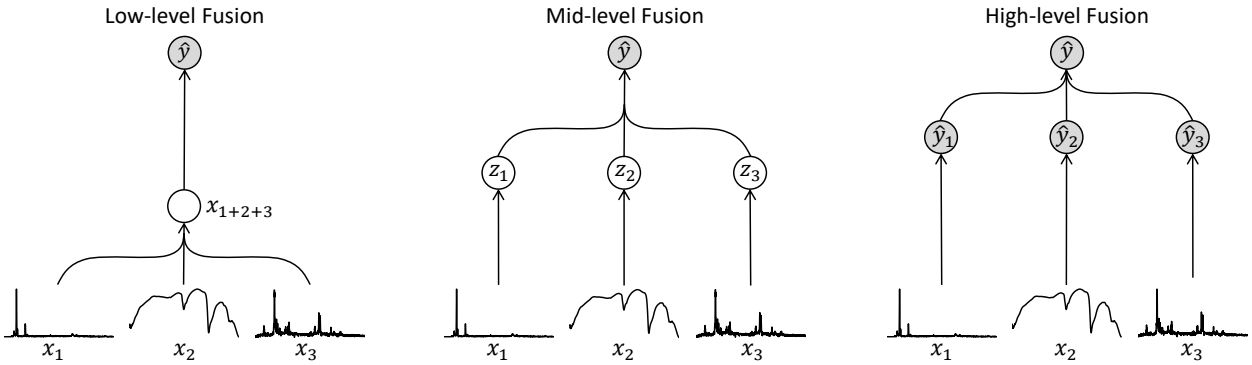


Figure 1: Sensor fusion models can by described as fitting into one of three levels, depicted here: low-level, mid-level, and high-level fusion (also known as data-level, intermediate or feature-level, and decision-level).

able selection or dimensionality reduction to each block prior to the fusion step. Using automatic hyperparameter optimization (HPO) [18] techniques, we optimized the architecture and hyperparameters of a convolutional neural network (CNN) for mid-level sensor fusion. We compare sensor fusion techniques on two prediction tasks individually: predicting lithium (Li) concentration and predicting zirconium (Zr) concentration using three types of spectra.

2. Materials and Methods

2.1. Data and Experiment Setup

This section provides details about the dataset, and the training and evaluation methodology. The dataset used in this study consists of 177 rock core samples, each about 8 cm in length, obtained from a lithium mineral exploration project. Three types of spectroscopy were used to analyze each sample: x-ray fluorescence (XRF), visible to short-wave infrared (Vis-NIR-SWIR) hyperspectral, and laser-induced breakdown spectroscopy (LIBS). Each spectrometer was used to scan each rock sample in four orientations, and the resulting spectra were averaged together. After obtaining spectral measurements, each sample was destructively analyzed using geochemical assays to determine ground truth.

As in many spectroscopy applications, resource limitations constrained the number of samples that could be studied. Each sample is manually moved into position for each sensor type and for each of 4 orientations, and a third-party assay lab was paid to perform geochemical analysis. Not only is this process time-consuming and expensive, but it also destroys the rock in the process, eliminating any possibility of further inquiry or studies.

2.1.1. X-ray Fluorescence

Energy-dispersive X-ray fluorescence (XRF) is a non-destructive analytical technique utilized for determining the elemental composition of a sample. When X-rays irradiate a sample, atoms absorb the energy and eject electrons, creating vacancies. These vacancies cause electrons from outer shells to transfer to inner shells and, consequently, characteristic X-rays are emitted with an energy equal to the difference between the two binding energies of the corresponding shells.

The XRF instrument utilized in this study uses a 50 kV X-ray tube with a silver (Ag) target and a detector that has a beryllium (Be) window. It generates a spectrum with 1024 channels from 2 keV to 33 keV. The spot size on the sample is around 2 cm in diameter. The unit is automatically moved to be approximately 10 cm away from the rock sample using a laser distance sensor.

2.1.2. Visible to Short-wave Infrared

A portable visible to short-wave infrared (Vis-NIR-SWIR) spectrometer was used to collect high-resolution spectral reflectance data in the range of 350 to 2500 nm with 1 nm intervals (2151 spectral channels). The instrument is an ASD TerraSpec Halo model from Malvern Panalytical that carried out auto-calibration. The technique is a contact method and the analysis spot size is approximately 2 to 3 cm in diameter.

2.1.3. Laser-Induced Breakdown Spectroscopy

Laser-induced breakdown spectroscopy (LIBS) is a technique that uses a high-energy laser pulse

to create a plasma on the sample surface, which emits light that can be analyzed by a spectrometer. Due to laser ablation, a small portion of the material is destroyed, but LIBS is still considered non-destructive because the spot size is around 0.2 mm in diameter, which is negligible compared to the mass of the whole rock. For each of the four rock sample orientations, 100 lasers pulses are produced, resulting in a total of 400 spectra that are averaged to produce one spectrum per rock sample.

The LIBS instrument is a FiberLIBS model from Secopta (Teltow, Germany) equipped with a 1064 nm Q-switched Nd:YAG laser that has a pulse duration of 1.5 ns, an energy output of 3 mJ, a maximum pulse rate of 100 Hz, and a spot size of around 0.2 mm. The spectrometer is a Czerny-Turner type with a CCD detector providing 2048 channels from 229.21 to 499.58 nm with a full width at half maximum (FWHM) of 0.20 to 0.5 nm.

2.1.4. Prediction Targets

Li and Zr were chosen as the prediction targets because they are relevant to the mineral exploration project where the rocks were collected and they represent examples where direct signals are expected (Zr) and not expected (Li) in the XRF spectrum. Neither element would have direct responses in Vis-NIR-SWIR. Both elements have the potential to show direct responses using LIBS. Since the sensors exhibit unique abilities, their responses are expected to complement each other in a sensor fusion model.

Ground truth Li and Zr concentrations are obtained by destructive geochemical assay providing gold-standard composition estimates. ALS, a commercial assay laboratory, carried out the analysis using their proprietary ME-MS61 and ME-XRF10 procedures [19]. The analysis uses 4-acid digestion followed by inductively coupled plasma mass spectrometry (ICP-MS). For Zr, 38 samples were above 500 ppm; these overlimit samples were subsequently analyzed by ALS's ME-XRF10 procedure. The samples were also accompanied by qualitative descriptions of the rock type, provided by a geologist during the drilling campaign.

2.1.5. Data Partitioning and Evaluation

Our evaluation and hyperparameter optimization method follows best practices [20, 21, 22, 23]. A test set of 35 samples (20% of the dataset) is selected using the DUPLEX algorithm. DUPLEX selection ensures that the two sets have the same diversity of samples [24]. Raw spectra from all three sensor types are concatenated and fed to the DUPLEX algorithm. By using the sensor data, the two sets will cover approximately the same region of sensor-data-space. Figure 3 shows the samples selected for the test set in red; for the purpose of visualization, Li% and Zr% concentrations are plotted.

We use 10-fold cross-validation (CV) with the remaining 142 samples to partition the data into a training set, T_k, and a validation set, V_k, for $k \in [1..10]$. The partitions are illustrated in Figure B.9.

An overview of model training, hyperparameter optimization (HPO), and evaluation is given in the pseudo-code in Figure 2. Root-mean-square error (RMSE) is used to score the models. Model hyperparameters are optimized to minimize the RMSE of cross-validation (RMSECV), which is the RMSE of the union of all validation sets (and is equivalent to minimizing the sum of squared errors summed over all folds, see Figure 2, line 6). The best model configuration from HPO (see line 7) is evaluated on the test set, and we report its error distribution as the RMSE of prediction (RMSEP) for each fold of cross-validation (calculated in line 11).

2.1.6. Spectral Pre-processing

Spectral data is often pre-processed prior to modeling. One of the most common methods is standardization using standard normal variate (SNV) [25]. SNV is applied to each spectrum by subtracting the mean of the spectrum and dividing by the standard deviation of the spectrum [26]. Although originally developed for near-infrared (NIR), it is a general technique that can also be applied to other types of spectra, such as LIBS [21, 27]. We test models with and without SNV applied to all three types of spectra.

A comparison of other pre-processing methods [28] is outside the scope of the present study. Pre-processing has been found [29] to reduce performance in at least one case. In preliminary testing, we found negligible difference in PLS model performance using the following pre-processing methods: the first and second derivative of the smoothed spectrum (smoothing via a Savitzky–Golay filter with window size of 13), and the first and second derivative of the smoothed SNV spectrum. The amount of smoothing (i.e. the window size) is a hyperparameter that ought to be tuned per sensor. To keep the hyperparameter search space small and to keep the number of experiments manageable,

these methods were excluded from the study. Future work is to consider the effect of pre-processing methods, and their combinations, on sensor fusion performance.

2.2. PLS-based Models

Three PLS-based sensor fusion models are tested based on partial least squares (PLS) [30] regression, which models the relationship between spectral data and the concentrations of Li and Zr in rock samples. Two hyperparameters are considered: the number of components (from 1 to 40) and whether to standardize the spectra with SNV. The hyperparameters are optimized through grid search to minimize the root mean square error of cross-validation (RMSECV). The decision of whether to use SNV in the sensor fusion models is decided by the validation results of the single-sensor PLS models.

2.2.1. Single-Sensor Models: with PLS

Individual PLS models are trained for each sensor type and each target. A grid search is performed to optimize the number of components from 1 to 40 and whether to use SNV. The best hyperparameter configuration is evaluated on the test set.

For Li, applying SNV to spectra in all sensor types produced a better RMSECV than using raw spectra. For Zr, the best approach was to use raw spectra from XRF and Vis-NIR-SWIR and to use SNV on LIBS. The PLS models that follow employ SNV based on these single-sensor results.

2.2.2. Low-level Sensor Fusion: with PLS

In low-level sensor fusion, the fusion is carried out at the data level (see fusion levels in Figure 1). This is achieved by concatenating spectra together then training a PLS model as usual (e.g., [1]). The predictions obtained in this way are equivalent to multiblock PLS (MB-PLS) algorithms [31]. We refer to this sensor fusion model as PLS-LL for "Partial Least Squares - Low Level." Whether to use raw or SNV spectra is pre-determined by the validation results of the single-sensor PLS models (from Section 2.2.1); SNV is applied prior to concatenation. However, the number of components (from 1 to 40) is again optimized through grid search.

2.2.3. High-level Sensor Fusion: with NNLS

In high-level, or decision-level, sensor fusion, fusion is achieved by combining "decisions," which are the predictions made by multiple trained models, in this case, the predicted Li or Zr concentrations from single-sensor PLS models. We call this sensor fusion model PLS-HL for "Partial Least Squares - High Level."

We use a non-negative least squares (NNLS) model to combine the prediction values. Lee et al. [4] found that this type of sensor fusion model outperformed MB-PLS. NNLS is a method used to solve linear least squares problems where the model coefficients are constrained to be positive [32]. The non-negative constraint aligns with our intuitions that the final prediction is essentially a weighted av-

```
function HPO(model)
    for config in configurations do
        SSE = 0 // Sum of squared errors
        for k in [1..10] do
            f_k = train(config, C_k) // Train model on training set C_k; see Figure B.9
            SSE += Σ_{<x,y>∈V_k} (f_k(x) − y)^2 // Score predictions of validation set V_k
    return f_{1..10} of config with lowest SSE // Note: RMSECV = √(SSE/142)
function final_evaluation(model)
    f_{1..10} = HPO(model)
    for k in [1..10] do
        RMSEP_k = Σ_{<x,y>∈T} √((y − f_k(x))^2 / |T|) // Score predictions of test set T
    boxplot(model, RMSEP_{1..10}) // Shown in Figure 7
```

Figure 2: Pseudo-code overview of model training, hyperparameter optimization (HPO function), and final evaluation. C_k, V_k, and T are sets of (x, y) pairs in training, validation, and test sets respectively (where k denotes k^{th} fold); these dataset partitions are shown in Figure B.9.

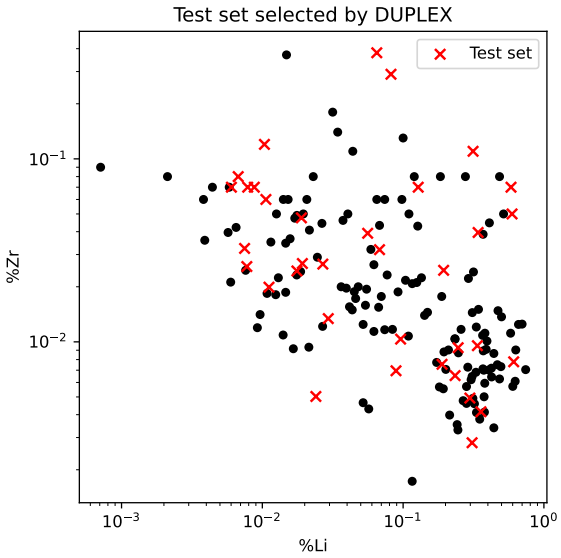


Figure 3: The DUPLEX algorithm selects a test set (red ×) that captures the same diversity of samples as the remaining samples.

erage of the predictions made by each single-sensor model. A negative weight, however, would indicate that an increase in the corresponding single-sensor model's predicted concentration is associated with a decrease in the final predicted concentration, which, intuitively, is not what we expect.

The NNLS model predictions are weighted sums of three predictions from the three best hyperparameter-tuned single-sensor PLS models. For each fold of cross-validation, the three single-sensor models make predictions on the training set (as per the data partitioning shown in Figure B.9), and then the NNLS model uses these predictions to calibrate its weights.

2.2.4. High-level Sensor Fusion: with ROSA

Response-Oriented Sequential Alternation (ROSA) [33, 34] is a state-of-the-art multiblock extension to PLS. In ROSA, components are constructed sequentially, using only one block per component. For each component, whichever block reduces the variance the most is selected. A benefit of this approach is that it is invariant to block scaling.

ROSA can be used for sensor fusion. When the input blocks to ROSA are spectra pre-processed in different ways, the result is a model that automatically chooses from the available pre-processing methods [35]; when the input blocks are spectra from different sensors, ROSA effectively performs sensor fusion. We use spectra from each of the three sensor types. Whether to use raw or SNV spectra is pre-determined by the validation results of the single-sensor PLS models (from Section 2.2.1). As in the other PLS models, the number of components (from 1 to 40) is optimized through grid search.

2.3. Neural Network Models

Two single-sensor neural networks (NN) and two sensor fusion models are developed. Each is developed in turn, as we freeze some hyperparameters after each one to reduce the size of the hyperparameter search space in subsequent models. The models are trained separately for each prediction target.

The training procedure for all the neural networks is based on standard practice [36, 37, 23]. Each input variable is standardized independently by subtracting the mean and dividing by the standard deviation [38]. The weights of the neural network are optimized by stochastic gradient descent using the ADAM [39] algorithm. All the layers (except the last) use exponential linear unit (ELU [40]) activations. Convolutional kernels and fully-connected weights are initialized by the He normal [41] initialization method. The batch size is set to the full size of the training data. The learning rate halves each time validation loss stops improving for 25 epochs, and early stopping is employed with a patience of 200 epochs; validation loss is calculated on 10% of training samples, selected randomly at the start of a new training run. Training runs for a maximum of 10000 epochs (but mean number of epochs was around 1300 epochs, due to early stopping).

2.3.1. Neural Network Hyperparameter Optimization

Automatic hyperparameter optimization is performed for all the neural network models following established protocols [42, 43]. There are many hyperparameters available to tune and testing all hyperparameter combinations is computationally prohibitive, even when only a subset is considered. To address this, an implementation of Bayesian optimization (BO), called hyperopt [44], is used to search through the space of hyperparameters to minimize root mean square error of cross-validation (RMSECV).

The BO algorithm chooses the configurations to test in line 2 of Figure 2. In the field of automated machine learning (Auto-ML), Bayesian optimization represents the current state-of-the-art for conducting a search through a hyperparameter search space. Bayesian optimization for HPO [45, 46, 47] employs a sequential approach to select hyperparameter configurations based on a combination of their estimated performance and the level of uncertainty associated with the estimate.

Our study focuses on a select number of hyperparameters, similar to previous studies [48, 23, 43], which are listed for each neural network model in the following sections. Each run of HPO was ran for about 700 trials. We ran this many because the best hyperparameter configuration stopped changing between 50 and 600 trials (but typically around 300), and then we ran each one longer until they all had at least 700 trials.

One challenge in HPO is that neural networks are known to produce different results each time they are trained [36], which adds noise to the objective. However, it has been demonstrated [43] that ensembling randomly-initialized neural networks is effective in reducing variance during hyperparameter optimization. The number of members in the ensemble can be selected to trade-off between variance and compute time. To determine the number of members, an analysis of the effect of ensemble size on variance was carried out (see Appendix A). This analysis was conducted using the initial configuration of the NN1 model, prior to hyperparameter optimization. Based on this analysis, variance is significantly reduced when the ensemble size reaches approximately 30 members. Beyond 30, both variance and mean RMSE show little sensitivity to the specific number chosen. We opted for a ensemble size of 40 members given the computational resources at our disposal.

2.3.2. NN1: Single-Sensor Neural Network

One-dimensional (1D) convolutional neural networks (CNNs) are a type of neural network commonly used for analyzing sequential data, such as spectra [49, 50, 51, 37, 52, 53, 54, 55, 56, 57, 58]. CNNs for spectral analysis all follow a similar pattern: one or more convolutional layers followed by one or more fully-connected layers, with an activation function after each layer. The convolution operation is useful for spectra because it automatically learns how to transform the spectra in the same way

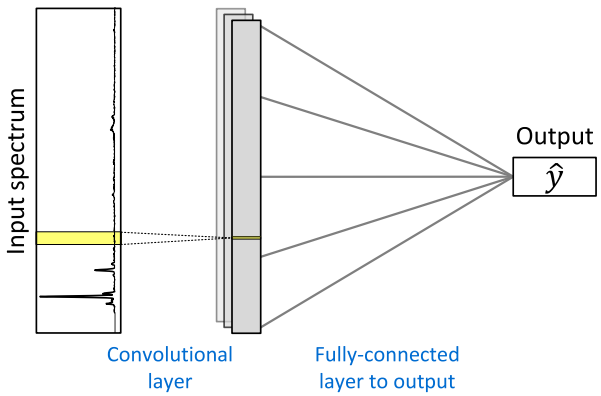


Figure 4: NN1 is a single-sensor model with one convolutional layer and one fully-connected layer which outputs a predicted concentration per input spectrum.

that pre-processing techniques do (such as smoothing and taking derivatives) [48, 37, 38, 59].

CNNs can take on many possible architectures; it is recommended [23] in chemometrics to start with simple networks and gradually increase complexity. Furthermore, it has been suggested [42] that, in chemometrics, a single convolutional layer may be sufficient, which is consistent with our own experience too. The first neural network (NN1) we test uses one convolutional layer and one fully-connected layer (also known as a dense layer), shown in Figure 4. The search space selected for NN1 is:

- Whether to standardize spectra using SNV
- L_2 regularization factor, which is a real number sampled uniformly from 10^{-4} to 100 in log space. The log space is useful because a similar impact is expected when changing the value from 0.001 to 0.002 as from 10 to 20, for example, whereas 10.001 to 10.002 is likely negligible.
- Learning rate (LR) (one of 1, 10^{-1}, or 10^{-2})
- Convolutional layer's filter size (one of 3, 5, 7, 11, 15, 21, 29, 41, 57, 79, 111, 155, or 217; search space is approximately logarithmic and rounded to the nearest odd number, ranging from 3 to 217)
- Convolutional layer's number of filters (an integer from 1 to 5)

HPO is carried out for all three sensor types and for both targets (Li and Zr). Additionally, whether

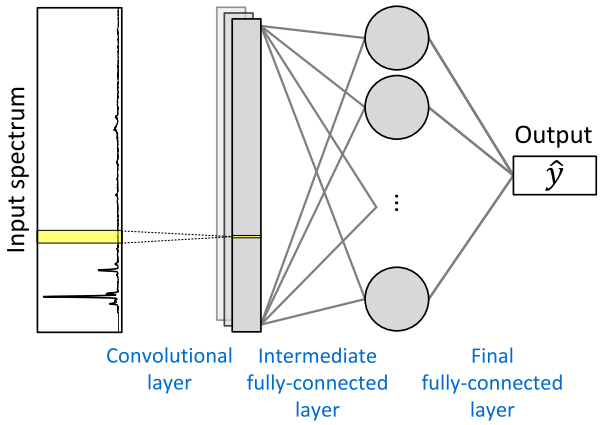


Figure 5: NN2 is similar to NN1 except an additional fully-connected layer is added.

to use SNV is optimized in a separate HPO run to allow analysis of the effect of standardization. Thus, 12 HPO runs are carried out for NN1.

2.3.3. High-level Sensor Fusion: with NNLS

Just like the PLS-HL model described in Section 2.2.3, the predictions from trained single-sensor neural networks may also be fused together to form a high-level sensor fusion model. We call this model NN1-HL.

2.3.4. NN2: Adding a hidden layer

The second neural network (NN2) we developed, shown in Figure 5, is similar to NN1 except an additional fully-connected layer is added (known as a hidden layer). This fully-connected layer is added because it will allow the sensor fusion neural network (described in the next section) to balance sensors via L_2 regularization. The additional layer also allows the model to learn more complex interactions and non-linearities, but at the risk of overfitting. While this could have been included as a hyperparameter in NN1, doing it separately allows us to evaluate the effect of adding this layer.

To reduce the search space, some hyperparameters are frozen based on the results of NN1's HPO (discussed in the results, Section 3.1, and Table 1). Since a fully-connected layer is added, the L_2 regularization factor should be re-optimized for NN2. Typically, the L_2 factor is the same among all fully-connected layers, but here they are separate because they will be optimized individually in the sensor fusion model. The HPO search space for NN2 is:

- L_2 regularization factor for the intermediate layer (a real number sampled uniformly from 10^{-4} to 100 in log space)
- L_2 regularization factor for the final layer (a real number sampled uniformly from 10^{-4} to 100 in log space)
- Number of units in the intermediate layer (an integer from 1 to 20)

2.3.5. NN3: Feature-level Sensor Fusion

The third neural network is a parallel-input convolutional neural network [15, 16] that we use to perform mid-level sensor fusion, which we call NN3-ML for "Neural Network 3 - Mid-Level." The model consists of layers arranged in parallel, each with its own input. Each input, known as a "block," is a set of spectra[1].

The neural network architecture is shown in Figure 6 and is composed of three convolutional layers, one per block, each with its own set of learned filter(s). Like NN2, a fully-connected layer follows each convolution layer; these fully-connected layers output intermediate features specific to each sensor. Finally, a fully-connected layer connects these intermediate features together—"fusing" the features from each sensor type—and outputs the predicted Li or Zr concentration. Instead of one shared regularization hyperparameter, each fully-connected layer has its own L_2 regularization factor hyperparameter:

- **L_2 on intermediate fully-connected layers**: The sensor-specific branches of the model are individually regularized to enable the sensor fusion neural network to balance the sensor types appropriately. The initial sampling distribution for each of these hyperparameters is based on the results of HPO from NN2. Specifically, a lognormal distribution centered at the optimal value from NN2 for each sensor and target, with a standard deviation such that 95% of samples fall within one order of magnitude.
- **L_2 on final fully-connected layer**: Due to an increase in the number of connections to the

[1] A "block" is multivariate data consisting of a set of related input variables. In this study, there are three blocks of spectra, one block per spectrometer.

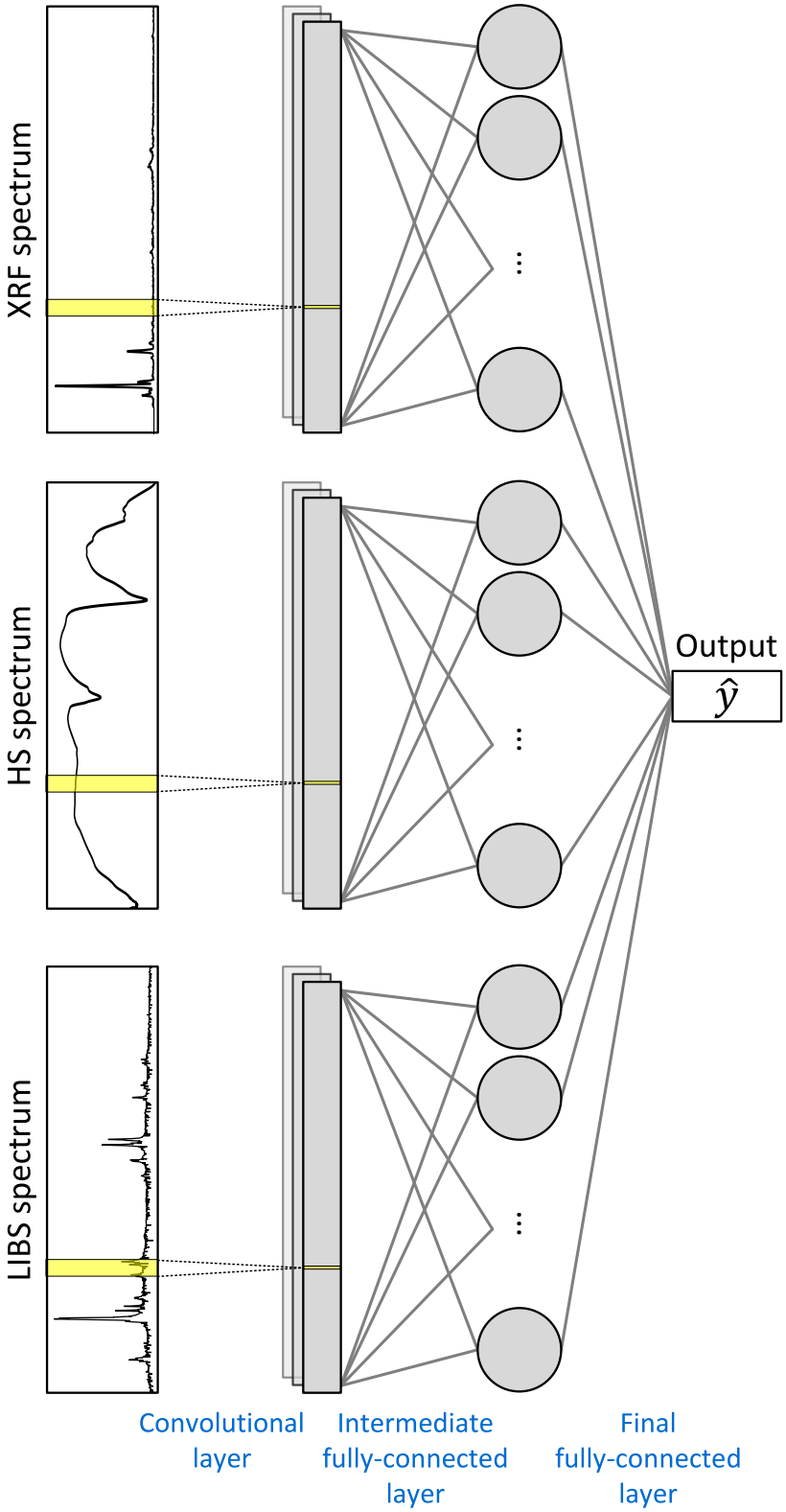


Figure 6: NN3 is essentially three of NN2 stacked in parallel, one per sensor (or "block"). The final fully-connected layer "fuses" the features from the intermediate layer together to make a prediction.

final output as compared to NN2, it is necessary to re-optimize the L_2 regularization factor. The initial sampling distribution in the search space is a lognormal distribution centered at the average value from NN2's HPO runs, with a standard deviation equal to that of the values from NN2's HPO runs.

2.4. Compute Hardware and Software

Source code used to train the models is available on GitHub[2]. Python 3.9, Octave 7.3, and R 4.2.2 were used in this study. We used the `DUPLEX` algorithm, which is available in the `prospectr` R package[3]. Neural network models were trained using `TensorFlow` 2.6[4] in Python. PLS models were trained using the `scikit-learn` Python package[5]. ROSA models were trained using the `multiblock` R package[6], which we ran via `rpy2`[7]. Experiments were run on a high-performance compute cluster called Cedar[8], which is hosted by Simon Fraser University for the Digital Research Alliance of Canada, with NVIDIA V100 and P100 GPUs and Intel Xeon CPUs (1352 GPU devices and 94528 CPU cores). A total of 374 GPU-days were utilized in training neural networks in the HPO experiments.

3. Results and Discussion

With numerous methods available in the literature for fusing data from multiple sensors, we trained models from each level and compared against single-sensor models for each sensor. The results of neural network hyperparameter optimization are reported first as these are utilized in constructing NN2 and NN3-ML. Then the results of single-sensor and sensor fusion models are presented.

3.1. Neural Network HPO

Some hyperparameters in NN2 are frozen based on the HPO results on NN1, listed in Table 1. Where possible, the same value is used across both targets for a given sensor (when the difference in prediction error is negligible); for instance, the best number of filters for the Vis-NIR-SWIR model targeting Zr was found to be 4, whereas targeting Li it was found to be 5 but 4 was almost as good, so 4 is used for both models in NN2.

The best hyperparameter values found from NN2 HPO are listed in Table 2. For NN3-ML, frozen hyperparameters include the ones previously frozen and also the best-found value for the number of hidden units (per sensor and per target) from NN2 HPO. The search space, described in Section 2.3.5, for L_2 regularization factors is based on the best L_2 values from NN2. The optimized hyperparameter values for NN3-ML are listed in Table 2.

More hyperparameter and architecture variations are possible, but are not necessary for testing sensor fusion and would greatly expand the hyperparameter search space if included. However, we conducted some preliminary testing (not shown) of some but found negligible differences in prediction accuracy on the validation set; the methods we tried were (1) batch normalization [60] after the convolutional layer or after the fully connected layers, (2) a proximity L_2 norm[9] [49], (3) smaller batch sizes [61], (4) learning rate scheduler hyperparameters (patience, reduction factor, and minimum learning rate), and (5) pre-processed spectra in various combinations (including first and second derivatives of smoothed spectra, for both PLS and neural networks). An open avenue of research is to establish which architecture(s) and hyperparameter configurations are best in chemometrics; narrowing down these options would greatly reduce the effort and compute time needed to develop neural network models.

3.1.1. Results of Single-Sensor Models

Results of the single-sensor models on two prediction tasks are presented in Figure 7. Each model's results is shown by a box representing the distribution of RMSEP scores from 10 trained instances of

[2] https://github.com/skylogic004/spectroscopy-neural-network-2

[3] https://rdrr.io/cran/prospectr/man/duplex.html and https://github.com/l-ramirez-lopez/prospectr

[4] https://www.tensorflow.org/versions/r2.6/api_docs/python/tf

[5] https://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html

[6] https://github.com/khliland/multiblock

[7] https://rpy2.github.io/

[8] https://docs.alliancecan.ca/wiki/Cedar

[9] The proximity L_2 norm introduces a regularizer on the convolution filters that encourages them to be smooth; in preliminary testing, we found that this produces nice smooth filters resembling Gaussian smoothing and first derivatives, which has the potential to be advantageous in some applications and may merit further investigation.

Target	Sensor	Standardization	LR	Conv. width	Num filters
Li Zr	XRF	SNV	0.01	111 41	1
Li Zr	Vis-NIR-SWIR	SNV	0.01	57	4
Li Zr	LIBS	SNV	0.01	57	3

Table 1: Frozen hyperparameters for NN2 as a result of NN1 HPO.

		Results of NN2 HPO			Results of NN3-ML HPO	
		Intermediate FC layer		Final FC	Sensor-specific FC	Final FC
Target	**Sensor**	**# units**	**L_2**	**L_2**	**L_2**	**L_2**
Li	XRF	6	2.26E-02	11.532	6.65E-03	10.608
	Vis-NIR-SWIR	19	9.18E-04	8.804	2.38E-04	
	LIBS	17	1.61E-03	3.420	4.36E-03	
Zr	XRF	9	2.48E-03	1.187	1.31E-03	30.250
	Vis-NIR-SWIR	8	5.02E-04	12.233	1.58E-04	
	LIBS	17	6.12E-03	9.936	6.11E-03	

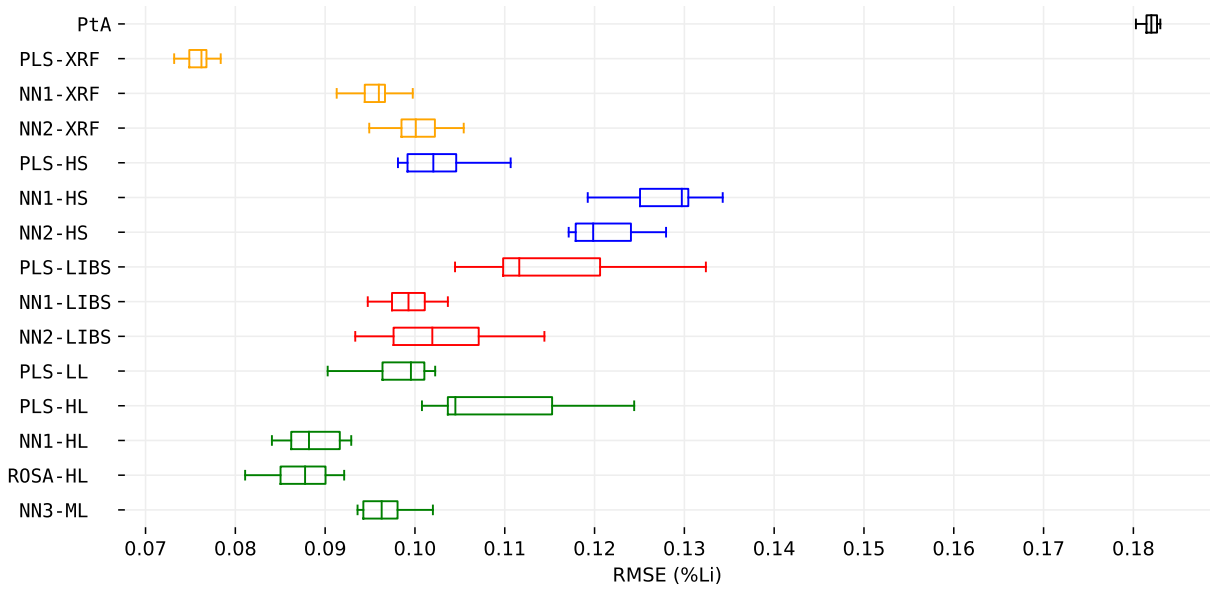
Table 2: Hyperparameter optimization (HPO) results for all fully-connected (FC) layers in NN1 and NN2

this model, where each instance was trained on a different fold of cross-validation. In order to facilitate interpretation, a naive baseline is included that outputs the average training set Li or Zr concentration (labeled "PtA" for "predict the average" and drawn as a black box in the figures). Single-sensor models are labeled with their sensor type appended to the name (e.g., NN1-XRF) and color-coded with XRF in orange, Vis-NIR-SWIR (labeled HS, for hyperspectral, for brevity) in blue, and LIBS in red. Each sensor has different strengths and weaknesses. Sensor-specific results are discussed next.

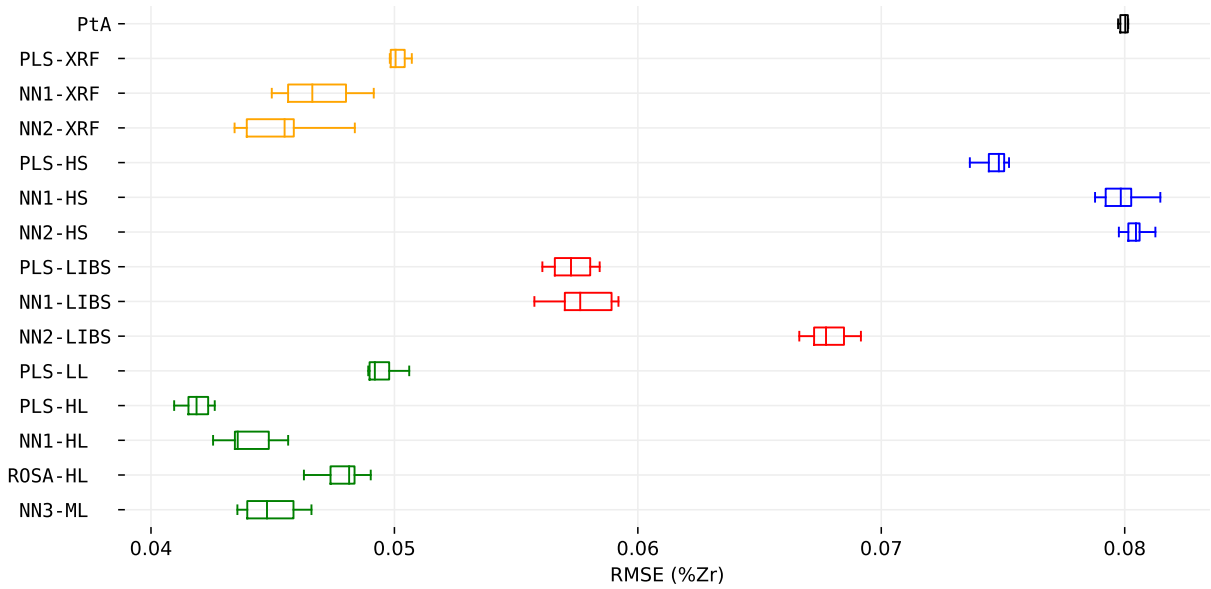
XRF: Li is not directly detectable by XRF, but a geochemical relationship was observed between Li and Rb which XRF is well-suited to detect. A strong Zr peak is expected in XRF. Models using the XRF sensor perform the best on average for both the Li and Zr target variables. While it was expected that Zr models using XRF would do well, Li performed better than expected despite relying on indirect signals.

Vis-NIR-SWIR: The Vis-NIR-SWIR models perform the poorest of the three sensors. Minerals that host Li are known to show spectral responses in the SWIR region, so it was expected that these models would exploit this. The host mineral for Zr (which is zircon) generally does not show spectral features. For Li, Vis-NIR-SWIR models were able to extract a signal, as expected, achieving much better performance than PtA. Zr models performed poorly, which is also as expected. Overall, Vis-NIR-SWIR models did worse than both the other sensor types.

LIBS: LIBS is expected to have difficulty with Li because the spectral range does not cover the key lines related to Li. The primary Li peak in LIBS is near 671 nm, which is outside of range, and the other Li peaks are small and only present in a few of the samples. Rb, which is correlated to Li, is also out of range. The Li models did very well considering that Li is out of range, especially with the neural network models (NN1 and NN2). Zr is within range in LIBS and the Zr models did reasonably well, albeit not as well as XRF. LIBS may not have done as well as XRF because of the difference in spot size for the LIBS system compared to the XRF and Vis-NIR-SWIR systems, which are considerably larger. Although the rocks analyzed are fine-grained and particles were not very heterogeneous, the scale of observation relative to the volume that was geochemically analyzed would have an impact on the representativity of the spectrometer response.



(a) Results of the lithium (Li) prediction task.



(b) Results of the zirconium (Zr) prediction task.

Figure 7: Test-set results of (a) Li and (b) Zr prediction tasks. Each box gives the distribution of test-set scores (measured in RMSE, which is in the same units as the target variable) obtained by training each model 10 times (on different folds of cross-validation). Lower is better. The edges of the boxes are the upper and lower quartiles of the distribution with a line at the median. The whiskers extend from the box to show the range of the data. Vis-NIR-SWIR hyperspectral is labeled HS for brevity. Model names are suffixed with the name of the sensor (XRF, HS, or LIBS) for single-sensor models or the sensor fusion level (LL: low level, ML: mid level, HL: high level) for models that use all three sensors. PtA means predict the average (which is a naive baseline).

3.1.2. Results of Sensor Fusion Models

Sensor fusion results are shown in Figure 7, drawn in green and labeled with the sensor fusion level (LL, ML, or HL) appended to the name (e.g., PLS-LL for PLS utilizing low-level fusion). The results indicate that the best sensor fusion model depends on the prediction target.

Li target: Of the sensor fusion models and with Li as the target, ROSA-HL obtains the best RMSE on average. The next best is the high-level fusion of neural network models (NN1-HL). Considering all the models for the Li prediction task (including single-sensor models and sensor fusion models), the results indicate that the single-sensor PLS model using XRF spectra achieves the lowest RMSE.

That the XRF-only approach performed best for Li was unexpected. While the XRF response was expected to exploit the relationship between Li and Rb, the Vis-NIR-SWIR data was expected to complement XRF in sensor fusion models because of the Li host minerals that display spectral responses in the SWIR region. Li predictions with Vis-NIR-SWIR and LIBS were also substantially better than PtA, which indicates that there are features in the reflectance (Vis-NIR-SWIR) and LIBS spectra related to the Li-mineralized rock type. Thus, it is surprising that the sensor fusion models were unable to leverage the different types of sensors to achieve better results than PLS with XRF alone.

Zr target: Of the sensor fusion models with Zr as the target, the best is PLS-HL, followed by NN1-HL, then NN3-ML. PLS-HL also outperforms all the single-sensor models, with a lower RMSE than any of the single-sensor PLS models, on which it is based. The improved performance may be because of the complementary information provided by the different sensor types.

The performance patterns for Zr are more in line with expected results. The XRF-only approaches were the best for the single sensors, which is logical since there is a strong XRF peak expected. The Vis-NIR-SWIR-only approach was marginally better than PtA, which is not surprising since it is difficult to discern Zr-enriched rocks visually, and the host mineral (zircon) generally does not show spectral features unless the field of view is dominated only by this mineral. Prediction error for LIBS fell between Vis-NIR-SWIR and XRF, which is expected since this sensor's response is compositionally controlled (which is good for predicting element concentrations) but the spot size is small (which reduces representativity). Sensor fusion approaches achieved as good and better results than any single sensor, thus demonstrating that integrating different types of sensor data can improve overall performances.

3.2. Interpreting Behavior of High-level Sensor Fusion

A benefit of high-level sensor fusion models is the ease of interpretability; NNLS weights are easily examined and ROSA reports the block order of the "winning" blocks.

NNLS weights for PLS-HL and NN1-HL are approximately equal across the 10 folds, so average values (rounded to 2 decimal places) are reported in Table 3. The PLS-HL model for Li performed poorly. From the weights, we observe that this model favors the PLS-LIBS model predictions the most. We investigated further and found that PLS-LIBS's training RMSE is much lower than its validation and test-set RMSE which indicates overfitting, and it fits the training data better than the other two sensors. As a result, PLS-HL for Li is weighing LIBS higher than XRF and Vis-NIR-SWIR, and is adopting a similar RMSE as the LIBS model on its own.

The remaining three NNLS models (PLS-HL for Zr and NN1-HL for Li and Zr) exhibit similar behaviors: XRF has the highest weight followed by LIBS, and the weight for Vis-NIR-SWIR is zero or almost zero. Low Vis-NIR-SWIR weight is consistent with the training and test-set predictions for Vis-NIR-SWIR which were worse than both XRF and LIBS. By not utilizing Vis-NIR-SWIR, these models are potentially missing out on complementary sensor information.

		XRF	**Vis-NIR-SWIR**	**LIBS**
PLS-HL	Li	0.05	0.07	0.91
	Zr	0.68	0.01	0.36
NN1-HL	Li	0.60	0.02	0.50
	Zr	0.84	0.00	0.30

Table 3: Model weights for PLS-HL and NN1-HL in the non-negative least squares (NNLS) sensor fusion model.

As for ROSA, in the Li prediction task it selects LIBS in the first component and XRF for the remaining 17 components; Vis-NIR-SWIR never won in any of the iterations. For the Zr prediction task, the ROSA model has 12 components. Results vary

among the 10 models from cross-validation, but to summarize, the components are mostly XRF except: in 4 models Vis-NIR-SWIR is used once (i.e., Vis-NIR-SWIR is used in 1 of 12 components), in 3 models LIBS is used once, and in 3 models neither Vis-NIR-SWIR nor LIBS is used (only XRF). Since the ROSA models are primarily picking XRF spectra in their components, it is not surprising that their RMSEP distribution closely matches the XRF models.

In summary, results indicate that XRF data is the most influential in the sensor fusion models, and that future work is needed to inject domain knowledge into these models in order to better exploit the other sensor types.

3.3. Modeling Considerations and Limitations

Aside from prediction error, other factors should be considered when choosing a sensor fusion method. The neural network models were the most time-consuming to develop. High-level fusion using NNLS is computationally fast but requires two steps: training individual models per sensor, then training the NNLS model using their outputs. In practice, individual models are often built anyway, so adding a high-level fusion model on top (such as PLS-HL or NN1-HL) is easy and practical, so long as all individual models perform well on their own; as we observed, when one model overfits then high-level fusion overfits as well. ROSA models are also quick to train, have only one hyperparameter (the number of components), and perform well. The main downside of PLS-based models is that further testing of pre-processing is typically required, whereas the neural network models likely do not benefit from it. NN1-HL may strike a good balance between the flexibility of neural networks and the simplicity of high-level sensor fusion.

Opportunity exists for a meta-analysis of the results. For example, it may be that certain clusters of samples have poor predictions, possibly related to other less quantitative variables such as rock type. Finding out why "difficult samples" perform poorly may lead to a better understanding of each sensor's and model's capabilities and limitations, such as when one sensor or model is applicable versus another. A meta-analysis may reveal patterns about where sensor fusion can have the greatest impact in the mineral exploration and mining industries, which are currently undergoing rapid digitalization. Additionally, there may be grade ranges that are more important than others. If the cut-off grade for a mine is 0.2% Li, then a sample with 0.6% Li predicted to have 0.7% Li is less relevant than a 0.1% Li sample predicted at 0.2% Li. In the first case, the rock is clearly of ore grade regardless of the prediction error. In the second case, the prediction error would result in the material being classified as ore when instead it should be treated as waste. While there was not one sensor fusion model that dominated the rest, having multiple models to choose from is useful when additional criteria arise in real-time sensing environments (e.g., sensor-based core logging and rock sorting).

Small datasets are an unfortunate reality because of the cost associated with sample collection and analysis. Training neural networks with limited data poses challenges compared to large datasets, due to the risk of overfitting given the number of parameters in a neural networks. In the literature, CNNs have been successful on various small datasets, including 80 NIR samples [62, 54], 192 Vis-NIR samples [63] 124 MIR samples [51], 219 NIR samples [64], and 60 NIR samples [65], among others [55]. An increase in data is expected to significantly enhance the performance of neural network sensor fusion models as compared to PLS models.

This paper serves as a preliminary investigation due to the reliance on a single dataset. More studies that test diverse domains are required before definitive recommendations can be made. Datasets with the ground truth and multiple sensors are typically expensive to create, so it is even more important to have multiple studies, so expensive decisions can be made with the best evidence.

4. Conclusion

While the neural network models were competitive, the most effective approach for sensor fusion was achieved through high-level PLS-based methods. The ROSA model for Li prediction and the PLS-HL model for Zr prediction yielded the best results among the sensor fusion models. We compared sensor fusion models—spanning low-level, mid-level, and high-level sensor fusion—to single-sensor models and found that sensor fusion improved predictions for one out of the two prediction tasks. For the Zr prediction task, combining information from three sensors resulted in the lowest RMSE on average, effectively leveraging the data from multiple sensors. For the Li prediction task, the PLS single-sensor model utilizing XRF

outperformed all the sensor fusion models. Among the single-sensor models, neural networks demonstrated better performance than PLS in two out of the six tasks. A parallel-input convolutional neural network was developed for sensor fusion. This model (NN3-ML) consistently achieved comparable RMSE to the models utilizing a single sensor (XRF), for both prediction tasks. It is important to recognize that improved performance cannot be guaranteed solely by increasing the number of sensor types or employing a more sophisticated prediction model. And, given the small size of the dataset in the experiments, larger datasets are likely to exhibit different behavior. Conducting multiple studies using diverse datasets is essential to gain deeper insights. This study should be seen as one of many contributing to a more comprehensive understanding of sensor fusion in spectroscopy.

Acknowledgements

We thank MineSense Technologies (https://minesense.com/) and Mitacs (https://www.mitacs.ca/) for supporting this research.

References

[1] A. Biancolillo, R. Bucci, A. L. Magrì, A. D. Magrì, F. Marini, Data-fusion for multiplatform characterization of an italian craft beer aimed at its authentication, Analytica Chimica Acta 820 (2014) 23–31. doi:https://doi.org/10.1016/j.aca.2014.02.024. URL https://www.sciencedirect.com/science/article/pii/S0003267014002220

[2] Q. Wang, J. Xiao, Y. Li, Y. Lu, J. Guo, Y. Tian, L. Ren, Mid-level data fusion of raman spectroscopy and laser-induced breakdown spectroscopy: Improving ores identification accuracy, Analytica Chimica Acta 1240 (2023) 340772. doi:https://doi.org/10.1016/j.aca.2022.340772. URL https://www.sciencedirect.com/science/article/pii/S0003267022013435

[3] P. M. Ramos, I. Ruisánchez, K. S. Andrikopoulos, Micro-raman and x-ray fluorescence spectroscopy data fusion for the classification of ochre pigments, Talanta 75 (4) (2008) 926–936. doi:https://doi.org/10.1016/j.talanta.2007.12.030. URL https://www.sciencedirect.com/science/article/pii/S0039914007008909

[4] H. W. Lee, A. Christie, J. Xu, S. Yoon, Data fusion-based assessment of raw materials in mammalian cell culture, Biotechnology and Bioengineering 109 (11) (2012) 2819–2828. doi:https://doi.org/10.1002/bit.24548. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bit.24548

[5] E. Borràs, J. Ferré, R. Boqué, M. Mestres, L. Aceña, O. Busto, Data fusion methodologies for food and beverage authentication and quality assessment – a review, Analytica Chimica Acta 891 (2015) 1–14. doi:https://doi.org/10.1016/j.aca.2015.04.042. URL https://www.sciencedirect.com/science/article/pii/S0003267015005528

[6] S. Fan, C. Li, W. Huang, L. Chen, Data fusion of two hyperspectral imaging systems with complementary spectral sensing ranges for blueberry bruising detection, Sensors 18 (12) (Mar 2018). doi:10.3390/s18124463. URL https://www.mdpi.com/1424-8220/18/12/4463

[7] V. Panchuk, I. Yaroshenko, A. Legin, V. Semenov, D. Kirsanov, Application of chemometric methods to XRF-data - a tutorial review, Analytica Chimica Acta 1040 (2018) 19 – 32. doi:https://doi.org/10.1016/j.aca.2018.05.023. URL http://www.sciencedirect.com/science/article/pii/S0003267018306123

[8] F. Marini, M. Tomassetti, M. Piacentini, L. Campanella, P. Flamini, Application of near infrared spectroscopy (NIR), x-ray fluorescence (XRF) and chemometrics to the differentiation of marmora samples from the mediterranean basin, Natural Product Research 33 (7) (2019) 1006–1014. doi:10.1080/14786419.2016.1190717. URL https://doi.org/10.1080/14786419.2016.1190717

[9] A. Maléchaux, Y. Le Dréau, J. Artaud, N. Dupuy, Control chart and data fusion for varietal origin discrimination: Application to olive oil, Talanta 217 (2020) 121115. doi:https://doi.org/10.1016/j.talanta.2020.121115. URL https://www.sciencedirect.com/science/article/pii/S0039914020304069

[10] P. Mishra, J. M. Roger, D. N. Rutledge, A. Biancolillo, F. Marini, A. Nordon, D. Jouan-Rimbaud-Bouveresse, MBA-GUI: A chemometric graphical user interface for multi-block data visualisation, regression, classification, variable selection and automated pre-processing, Chemometrics and Intelligent Laboratory Systems 205 (2020) 104139. doi:https://doi.org/10.1016/j.chemolab.2020.104139. URL https://www.sciencedirect.com/science/article/pii/S0169743920303956

[11] P. Mishra, F. Marini, B. Brouwer, J. M. Roger, A. Biancolillo, E. Woltering, E. H. van Echtelt, Sequential fusion of information from two portable spectrometers for improved prediction of moisture and soluble solids content in pear fruit, Talanta 223 (2021) 121733. doi:https://doi.org/10.1016/j.talanta.2020.121733. URL https://www.sciencedirect.com/science/article/pii/S0039914020310249

[12] S. M. Azcarate, R. Ríos-Reina, J. M. Amigo, H. C. Goicoechea, Data handling in data fusion: Methodologies and applications, TrAC Trends in Analytical Chemistry 143 (2021) 116355. doi:https://doi.org/10.1016/j.trac.2021.116355. URL https://www.sciencedirect.com/science/article/pii/S0165993621001783

[13] P. Mishra, J.-M. Roger, D. Jouan-Rimbaud-Bouveresse, A. Biancolillo, F. Marini, A. Nordon, D. N. Rutledge, Recent trends in multi-block data analysis in chemometrics for multi-source data integration, TrAC Trends in Analytical Chemistry 137 (2021) 116206.

doi:https://doi.org/10.1016/j.trac.2021.116206.
URL https://www.sciencedirect.com/science/article/pii/S0165993621000285

[14] S. D. Brown, R. Tauler, B. Walczak, Comprehensive chemometrics: chemical and biochemical data analysis, Elsevier, 2020.

[15] Puneet Mishra, D. Passos, Deep multiblock predictive modelling using parallel input convolutional neural networks, Analytica Chimica Acta 1163 (2021) 338520. doi:https://doi.org/10.1016/j.aca.2021.338520.
URL https://www.sciencedirect.com/science/article/pii/S0003267021003469

[16] A. Jenul, S. Schrunner, B. N. Huynh, R. Helin, C. M. Futsæther, K. H. Liland, O. Tomic, Ranking feature-block importance in artificial multiblock neural networks, in: E. Pimenidis, P. Angelov, C. Jayne, A. Papaleonidas, M. Aydin (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2022, Springer Nature Switzerland, Cham, 2022, pp. 163–175.

[17] A. Ng, The rise of end-to-end learning, in: Machine learning yearning, 2018, pp. 91–92.
URL https://info.deeplearning.ai/machine-learning-yearning-book

[18] M. Feurer, F. Hutter, Hyperparameter optimization, in: F. Hutter, L. Kotthoff, J. Vanschoren (Eds.), Automatic Machine Learning: Methods, Systems, Challenges, Springer, 2019, pp. 3–38.
URL https://www.automl.org/book/

[19] ALS, Four acid digest & advanced icp-ms technology, https://www.alsglobal.com/-/media/ALSGlobal/Resources-Grid/ALS_4AcidsDigest_web.pdf, accessed: 2023-01-11 (2023).

[20] K. H. Esbensen, P. Geladi, Principles of proper validation: use and abuse of re-sampling for validation, Journal of Chemometrics 24 (3-4) (2010) 168–187. doi:https://doi.org/10.1002/cem.1310.
URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.1310

[21] J. El Haddad, L. Canioni, B. Bousquet, Good practices in LIBS analysis: Review and advices, Spectrochimica Acta Part B: Atomic Spectroscopy 101 (2014) 171–182. doi:https://doi.org/10.1016/j.sab.2014.08.039.
URL https://www.sciencedirect.com/science/article/pii/S0584854714002158

[22] F. Westad, F. Marini, Validation of chemometric models – a tutorial, Analytica Chimica Acta 893 (2015) 14–24. doi:https://doi.org/10.1016/j.aca.2015.06.056.
URL https://www.sciencedirect.com/science/article/pii/S0003267015008922

[23] D. Passos, P. Mishra, A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks, Chemometrics and Intelligent Laboratory Systems 223 (2022) 104520. doi:https://doi.org/10.1016/j.chemolab.2022.104520.
URL https://www.sciencedirect.com/science/article/pii/S0169743922000314

[24] R. D. Snee, Validation of regression models: Methods and examples, Technometrics 19 (4) (1977) 415–428.
URL http://www.jstor.org/stable/1267881

[25] R. J. Barnes, M. S. Dhanoa, S. J. Lister, Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra, Applied Spectroscopy 43 (5) (1989) 772–777. doi:10.1366/0003702894202201.
URL https://doi.org/10.1366/0003702894202201

[26] M. Zeaiter, J.-M. Roger, V. Bellon-Maurel, Robustness of models developed by multivariate calibration. part ii: The influence of pre-processing methods, TrAC Trends in Analytical Chemistry 24 (5) (2005) 437–445. doi:https://doi.org/10.1016/j.trac.2004.11.023.
URL https://www.sciencedirect.com/science/article/pii/S0165993605000580

[27] P. Pořízka, J. Klus, A. Hrdlička, J. Vrábel, P. Škarková, D. Prochazka, J. Novotný, K. Novotný, J. Kaiser, Impact of laser-induced breakdown spectroscopy data normalization on multivariate classification accuracy, J. Anal. At. Spectrom. 32 (2017) 277–288. doi:10.1039/C6JA00322B.
URL http://dx.doi.org/10.1039/C6JA00322B

[28] M. P. Campos, M. S. Reis, Data preprocessing for multiblock modelling – a systematization with new methods, Chemometrics and Intelligent Laboratory Systems 199 (2020) 103959. doi:https://doi.org/10.1016/j.chemolab.2020.103959.
URL https://www.sciencedirect.com/science/article/pii/S0169743919305350

[29] P. Mishra, D. N. Rutledge, J.-M. Roger, K. Wali, H. A. Khan, Chemometric pre-processing can negatively affect the performance of near-infrared spectroscopy models for fruit quality prediction, Talanta 229 (2021) 122303. doi:https://doi.org/10.1016/j.talanta.2021.122303.
URL https://www.sciencedirect.com/science/article/pii/S0039914021002241

[30] H. A. O. Wold, Estimation of principal components and related models by iterative least squares, Multivariate analysis (1966) 391–420.
URL https://ci.nii.ac.jp/naid/20001378860/en/

[31] A. Baum, L. Vermue, Multiblock PLS: Block dependent prediction modeling for python, Journal of Open Source Software 4 (34) (2019) 1190. doi:10.21105/joss.01190.
URL https://doi.org/10.21105/joss.01190

[32] C. L. Lawson, R. J. Hanson, Linear least squares with linear inequality constraints, in: Solving Least Squares Problems, Society for Industrial and Applied Mathematics, 1995, Ch. 23, pp. 158–173. doi:10.1137/1.9781611971217.
URL https://epubs.siam.org/doi/abs/10.1137/1.9781611971217.ch23

[33] K. H. Liland, T. Næs, U. G. Indahl, ROSA—a fast extension of partial least squares regression for multiblock data analysis, Journal of Chemometrics 30 (11) (2016) 651–662. doi:https://doi.org/10.1002/cem.2824.
URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.2824

[34] A. K. Smilde, T. Næs, K. H. Liland, Supervised Methods, John Wiley & Sons, Ltd, 2022, Ch. 7, pp. 187–230. doi:https://doi.org/10.1002/9781119600978.ch7.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119600978.ch7

[35] P. Mishra, J. M. Roger, F. Marini, A. Biancolillo, D. N. Rutledge, Pre-processing ensembles with response oriented sequential alternation calibration (PROSAC): A step towards ending the pre-processing search and optimization quest for near-infrared spectral modelling, Chemometrics and Intelligent

Laboratory Systems 222 (2022) 104497. doi:https://doi.org/10.1016/j.chemolab.2022.104497.
URL https://www.sciencedirect.com/science/article/pii/S0169743922000089

[36] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: G. Montavon, G. B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade: Second Edition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 437–478. doi:10.1007/978-3-642-35289-8_26.

[37] C. Cui, T. Fearn, Modern practical convolutional neural networks for multivariate regression: Applications to NIR calibration, Chemometrics and Intelligent Laboratory Systems 182 (2018) 9–20. doi:10.1016/j.chemolab.2018.07.008.

[38] R. Helin, U. G. Indahl, O. Tomic, K. H. Liland, On the possible benefits of deep learning for spectral preprocessing, Journal of Chemometrics 36 (2) (2022) e3374. doi:https://doi.org/10.1002/cem.3374.
URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.3374

[39] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[40] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), arXiv preprint arXiv:1511.07289 (2015).
URL https://arxiv.org/abs/1511.07289

[41] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification (2015). arXiv:1502.01852.

[42] P. Mishra, D. Passos, Multi-output 1-dimensional convolutional neural networks for simultaneous prediction of different traits of fruit based on near-infrared spectroscopy, Postharvest Biology and Technology 183 (2022) 111741. doi:https://doi.org/10.1016/j.postharvbio.2021.111741.
URL https://www.sciencedirect.com/science/article/pii/S0925521421002805

[43] M. Dirks, D. Poole, Automatic neural network hyperparameter optimization for extrapolation: Lessons learned from visible and near-infrared spectroscopy of mango fruit, Chemometrics and Intelligent Laboratory Systems 231 (Dec 2022). doi:https://doi.org/10.1016/j.chemolab.2022.104685.
URL https://arxiv.org/abs/2210.01124

[44] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, Vol. 28 of Proceedings of Machine Learning Research, PMLR, Atlanta, Georgia, USA, 2013, pp. 115–123.
URL https://proceedings.mlr.press/v28/bergstra13.html

[45] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: a python library for model selection and hyperparameter optimization, Computational Science and Discovery 8 (1) (2015) 014008. doi:10.1088/1749-4699/8/1/014008.
URL https://doi.org/10.1088/1749-4699/8/1/014008

[46] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (1) (2016) 148–175. doi:10.1109/JPROC.2015.2494218.

[47] D. Passos, P. Mishra, An automated deep learning pipeline based on advanced optimisations for leveraging spectral classification modelling, Chemometrics and Intelligent Laboratory Systems 215 (2021) 104354. doi:https://doi.org/10.1016/j.chemolab.2021.104354.
URL https://www.sciencedirect.com/science/article/pii/S0169743921001222

[48] E. J. Bjerrum, M. Glahder, T. Skov, Data augmentation of spectral data for convolutional neural network (CNN) based deep chemometrics, CoRR abs/1710.01927 (2017).
URL http://arxiv.org/abs/1710.01927

[49] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T. N. Tran, L. M. Buydens, E. Marchiori, Convolutional neural networks for vibrational spectroscopic data analysis, Analytica Chimica Acta 954 (2017) 22 – 31. doi:https://doi.org/10.1016/j.aca.2016.12.010.
URL http://www.sciencedirect.com/science/article/pii/S0003267016314842

[50] J. Liu, M. Osadchy, L. Ashton, M. Foster, C. J. Solomon, S. J. Gibson, Deep convolutional neural networks for raman spectrum recognition: a unified solution, Analyst 142 (2017) 4067–4074. doi:10.1039/C7AN01371J.
URL http://dx.doi.org/10.1039/C7AN01371J

[51] S. Malek, F. Melgani, Y. Bazi, One-dimensional convolutional neural networks for spectroscopic signal regression, Journal of Chemometrics 32 (5) (2018) e2977, e2977 CEM-17-0106.R1. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.2977, doi:10.1002/cem.2977.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.2977

[52] X. Fan, W. Ming, H. Zeng, Z. Zhang, H. Lu, Deep learning-based component identification for the raman spectra of mixtures, Analyst 144 (2019) 1789–1798. doi:10.1039/C8AN02212G.
URL http://dx.doi.org/10.1039/C8AN02212G

[53] M. Chatzidakis, G. A. Botton, Towards calibration-invariant spectroscopy using deep learning, Scientific Reports 9 (1) (2019) 2126. doi:10.1038/s41598-019-38482-1.
URL https://doi.org/10.1038/s41598-019-38482-1

[54] X. Zhang, T. Lin, J. Xu, X. Luo, Y. Ying, DeepSpectra: An end-to-end deep learning approach for quantitative spectral analysis, Analytica Chimica Acta 1058 (2019) 48 – 57. doi:https://doi.org/10.1016/j.aca.2019.01.002.
URL http://www.sciencedirect.com/science/article/pii/S0003267019300169

[55] J. Yang, J. Xu, X. Zhang, C. Wu, T. Lin, Y. Ying, Deep learning for vibrational spectral analysis: Recent progress and a practical guide, Analytica Chimica Acta (2019). doi:https://doi.org/10.1016/j.aca.2019.06.012.
URL http://www.sciencedirect.com/science/article/pii/S0003267019307342

[56] P. Mishra, D. Passos, A synergistic use of chemometrics and deep learning improved the predictive performance of near-infrared spectroscopy models for dry matter prediction in mango fruit, Chemometrics and Intelligent Laboratory Systems 212 (2021) 104287. doi:https://doi.org/10.1016/j.chemolab.2021.104287.

URL https://www.sciencedirect.com/science/article/pii/S0169743921000551

[57] K. A. Einarson, A. Baum, T. B. Olsen, J. Larsen, I. Armagan, P. A. Santacoloma, L. K. H. Clemmensen, Predicting pectin performance strength using near-infrared spectroscopic data: A comparative evaluation of 1-D convolutional neural network, partial least squares, and ridge regression modeling, Journal of Chemometrics 36 (2) (2022) e3348. doi:https://doi.org/10.1002/cem.3348.
URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.3348

[58] G. Alix, E. Lymer, G. Zhang, M. Daly, X. Gao, A comparative performance of machine learning algorithms on laser-induced breakdown spectroscopy data of minerals, Journal of Chemometrics (2022). doi:https://doi.org/10.1002/cem.3400.
URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.3400

[59] G. B. Rossi, V. A. Lozano, A. C. Olivieri, Spectral pre-processing and non-linear calibration with convolutional kernel partial least-squares. teaching new tricks to an old dog, Chemometrics and Intelligent Laboratory Systems 233 (2023) 104736. doi:https://doi.org/10.1016/j.chemolab.2022.104736.
URL https://www.sciencedirect.com/science/article/pii/S0169743922002477

[60] D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, ArXiv abs/1804.07612 (2018).

[61] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, G. E. Dahl, Measuring the effects of data parallelism on neural network training (2019). arXiv:1811.03600.

[62] C. Yuanyuan, W. Zhibin, Quantitative analysis modeling of infrared spectroscopy based on ensemble convolutional neural networks, Chemometrics and Intelligent Laboratory Systems 181 (2018) 1 – 10. doi:https://doi.org/10.1016/j.chemolab.2018.08.001.
URL http://www.sciencedirect.com/science/article/pii/S0169743918301904

[63] X. Yu, H. Lu, Q. Liu, Deep-learning-based regression model and hyperspectral imaging for rapid detection of nitrogen concentration in oilseed rape (brassica napus l.) leaf, Chemometrics and Intelligent Laboratory Systems 172 (2018) 188–193. doi:https://doi.org/10.1016/j.chemolab.2017.12.010.
URL https://www.sciencedirect.com/science/article/pii/S0169743917306780

[64] C. Ni, D. Wang, Y. Tao, Variable weighted convolutional neural network for the nitrogen content quantization of masson pine seedling leaves with near-infrared spectroscopy, Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy 209 (2019) 32–39. doi:https://doi.org/10.1016/j.saa.2018.10.028.
URL https://www.sciencedirect.com/science/article/pii/S1386142518309557

[65] Y. yuan Chen, Z. bin Wang, Feature selection based convolutional neural network pruning and its application in calibration modeling for nir spectroscopy, Chemometrics and Intelligent Laboratory Systems 191 (2019) 103–108. doi:https://doi.org/10.1016/j.chemolab.2019.06.004.
URL https://www.sciencedirect.com/science/article/pii/S0169743919302126

Appendix A. Ensemble Size

To choose the number of members in the ensemble, an analysis was conducted to examine the relationship between RMSE and ensemble size prior to the experiments described in this paper. A CNN like NN1 was used for this analysis, before hyperparameter optimization was conducted.

Following the approach used previously [43], the RMSE distributions in Figure A.8 are computed by training the neural network 5000 times and simulating ensembles of different sizes. Each ensemble is a random, but unique, combination of N models. We generate 200 ensembles of each size, N, and plot the distribution of RMSE scores. From these results, it's apparent that the variance is significantly reduced after around $N = 30$, with diminishing returns as N increases further. It is important to point out that the number of members should be adjusted according to the requirements of the prediction task and the computational resources available to the practitioner.

Appendix B. Data Partitioning

Visualization of the data splitting procedure is shown in Figure B.9, with details provided in Section 2.1.5.

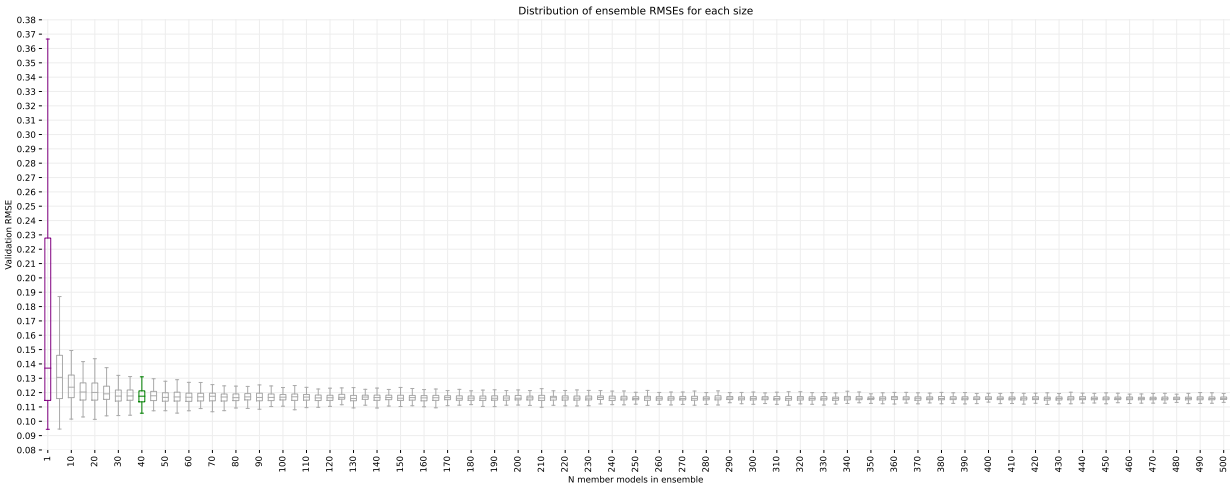


Figure A.8: Boxplot showing how much RMSE decreases as the size of the ensemble increases. Size $N = 1$ (purple) indicates the distribution of RMSE scores over 5000 single re-trainings (with random initialization) of the neural network model. Each box (where $N > 1$) is a distribution over 200 simulated ensembles, where each ensemble is formed from a unique combination of N trained models from the pre-computed set of 5000. We use ensembles with 40 member models in our experiments (green).

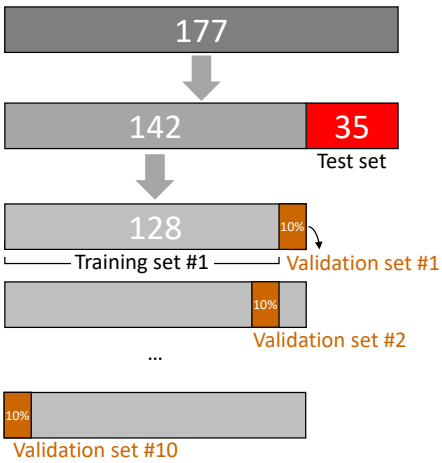


Figure B.9: The dataset contains 177 rock samples split into one test set, 10 training sets, and 10 validation sets for cross-validation.