Improving XRD Analysis with Machine Learning

Rachel E. Drapeau

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Barry R. Bickmore, Chair
Emily J. Evans
Stephen T. Nelson

Department of Geological Sciences

Brigham Young University

ABSTRACT

Improving XRD Analysis with Machine Learning

Rachel E. Drapeau
Department of Geological Sciences, BYU
Master of Science

X-ray diffraction analysis (XRD) is an inexpensive method to quantify the relative proportions of mineral phases in a rock or soil sample. However, the analytical software available for XRD requires extensive user input to choose phases to include in the analysis. Consequently, analysis accuracy depends greatly on the experience of the analyst, especially as the number of phases in a sample increases (Raven & Self, 2017; Omotoso, 2006). The purpose of this project is to test whether incorporating machine learning methods into XRD software can improve the accuracy of analyses by assisting in the phase-picking process. In order to provide a large enough sample of X-ray diffraction (XRD) patterns and their known compositions to train the machine learning models, I created a dataset of 1.5 million calculated XRD patterns of realistic mineral mixtures. These synthetic XRD patterns were calculated using crystal structure files from the American Mineralogist Crystal Structure Database (AMCSD) with mineral occurrence data from the Mineral Evolution Database (MED) to mimic geologic knowledge used by expert analysts. Using this dataset, I trained and refined a variety of machine learning models to determine which model is most accurate in identifying the correct mineral phases.

Keywords: X-ray diffraction analysis, XRD, machine learning, Rietveld method, crystal structure, classification, decision trees, bagged decision trees, data generation, mineral, mixture

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my husband, Joseph. I would never have completed my degree without his loving support and encouragement. He was the one who would help me out of my slumps and provide invaluable insight when problems occurred. I am immensely grateful and so lucky to have such a loving and selfless spouse. Thank you, and our children, for being my light and my motivation.

Thank you to my advisor, Dr. Barry Bickmore, for all of his support and flexibility. I really appreciate his understanding when trying to balance my master's program with raising my family. I'm very grateful he gave me this opportunity and for his continued support.

I would also like to thank Dr. Emily Evans for her assistance and advisement on this project. Her machine learning and math expertise was invaluable. Thank you to Dr. Stephen Nelson as well for his input and insight into the practicality of this project as well.

Thank you to Steve Maroney in the Mathematics Department for his assistance with the computer servers I used to generate the data and train the models.

Many thanks to Karla Ward and Dr. John McBride for helping me with the administrative hiccups I came across and their support and encouragement.

Thank you to my family and friends for their support with our family and vast encouragement as I worked on my thesis. They helped get me through the home stretch and I'm lucky to have so many people on my side.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Geological studies generally require analysis of the minerals present in rock, soil, and sediment samples to provide insight about conditions of formation. There are several available analytical techniques, such as X-ray diffraction (XRD), X-ray fluorescence (XRF), inductively coupled plasma mass spectrometry (ICP-MS), scanning electron microscopy (SEM), and electron microprobe, but they vary widely in the amount of information they provide to help infer the minerals' identities, the spatial range of analysis, and the cost. Quantitative X-ray diffraction analysis is an inexpensive method that can quantify the relative amounts of phases present in an entire sample. As such, it has become a popular method to determine the minerals present and their abundances.

A powder XRD instrument detects the angles at which X-rays diffract when interacting with a powdered, crystalline sample by utilizing Bragg's law (see Section 2.2.2). Those angles are determined by the length of unit cell repeat spacings in the crystal structures of the phases

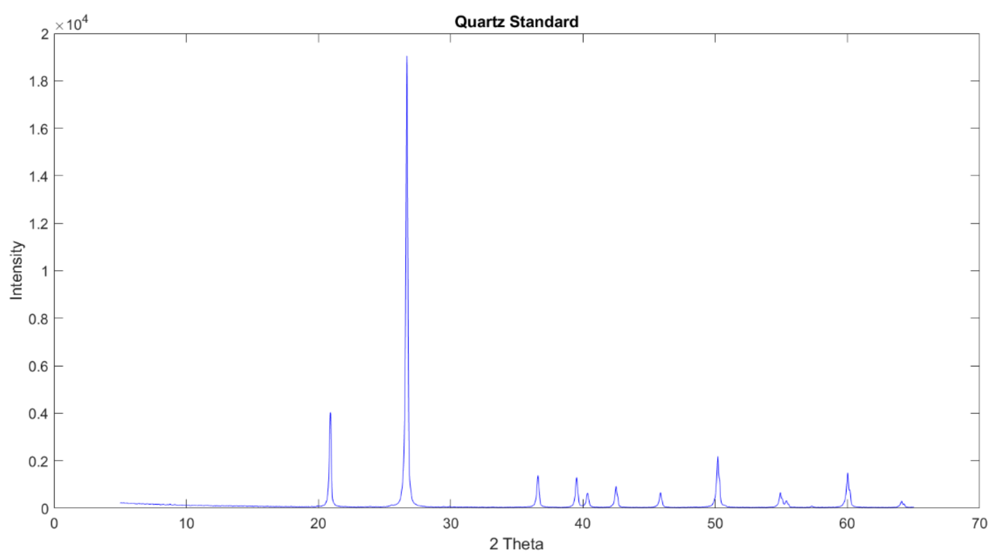

Figure 1. XRD Pattern for quartz using a copper anode X-ray tube.
The two highest intensity peaks at 20.88° and 26.66° 2θ are highly diagnostic of quartz, even in some mixtures. These peaks correspond to crystal structure repeat spacings of 4.255 and 3.343 respectively.

present. The relative intensities of the peaks depend both on the crystal structures and the identities of the atoms present in different sites in the structures (Pecharsky & Zavalij, 2009, p. 151, 161). Figure 1 shows an example powder XRD pattern of the mineral quartz (SiO_2). This pattern of peaks is highly diagnostic for single-phase samples, as it is easy to match the peak pattern to the patterns produced by single-phase standards, or even calculated from known crystal structures. However, if a sample contains multiple phases, the pattern must be modeled through XRD as a linear combination of single-phase patterns, which is then used to determine the amount of each phase present in the sample. This makes identification of the different phases in the sample difficult, because multiple phases can have major peaks at the same angles. If a sample contains an unknown number of phases, there can be many combinations that could possibly explain the positions and relative sizes of the peaks present.

The problem of identifying multiple phases correctly in XRD analysis has inspired research on the best practices for obtaining accurate results. One such project is the Reynolds Cup hosted by the Clay Minerals Society. Analysts in the competition perform their preferred method of quantitative phase analysis on a synthetic mixture, and then submit their results along with descriptions of their analytical procedures. Researchers then compare the methods to the accuracy of the results from each analyst. In past research, Raven & Self (2017) and Omotoso (2006) discovered that the most important factor in accurate quantitative phase analysis is the experience of the analyst rather than the method used. This is because the analyst must tell the analysis software which mineral phases to include for XRD fitting from a much larger set of possibilities.

The only guidance usually given by the software for picking candidate phases is a simple peak-matching routine, in which a list of the phases whose major peaks match major peaks in the sample pattern is given. In order to produce an accurate analysis, the analyst must use their geologic knowledge of the sample to review and narrow down the list of candidate phases for the software to optimize. This is usually done iteratively, until the analyst is satisfied with the match between the sample pattern and the pattern calculated from the selected phases. Novice analysts tend to treat the analytical software as a black box, accepting rare minerals or phases that are geologically or geochemically incompatible with the sample into their analysis (Raven & Self, 2017). On the other hand, expert analysts tend to incorporate other geologic knowledge, such as mineral prevalence and co-occurrence, when deciding which phases to include, thus resulting in more accurate analyses.

To alleviate this problem, I attempt to narrow the gap between novice and expert XRD analysts by training machine learning algorithms to accurately determine what phases are present in XRD analysis. Machine learning is a type of artificial intelligence that is used to address problems that cannot be solved using a straightforward algorithm. In other words, the problem cannot be solved easily by a step-by-step set of instructions for the computer without additional user input. Machine learning uses various statistical methods and algorithm types to mimic choices that human users might make to produce the best results. Unsupervised machine learning techniques identify patterns in data that are difficult for humans to recognize due to the data's complexity. Supervised machine learning techniques are used to create models to predict some feature of interest, or target, in the data. In this project, the target is the classification of the mineral phases in XRD patterns. Given enough high-quality training data, supervised machine learning is often able to create effective predictive models.

Using these predictive models and training data to mimic expert knowledge, machine learning has the potential to automate accurate analyses by assisting in the phase-picking process. However, these models are only as good as the data from which the machine learning is given to train. In order to produce accurate and generalizable predictive models, machine learning algorithms need large amounts of training data, on the order of thousands of samples. Such a database of experimentally derived XRD patterns with known compositions does not currently exist and would be extremely time-consuming and expensive to create. Therefore, I developed a synthetic database of realistic mineral mixtures derived from the American Mineralogist Crystal Structure Database (Downs, 2003), the Mineral Evolution Database (MED), MATLAB code, and the Rietveld equation.

Using this generated data set, I trained various machine learning models to be able to identify whether a target phase was included in a sample. Using the most consistently accurate of these models, I trained models of the same type for 72 different phases and tested their results on the XRD patterns of real mixtures. Using feature ranking algorithms, I also gained insight into helpful features of an XRD pattern that are most important to correctly identifying certain phases. These models and insights can aid novice analysts to produce more accurate XRD analyses.

2. BACKGROUND

To understand the methods discussed, more detailed explanations of XRD analysis and machine learning are provided in this section. Although there are many methods for quantitative phase analysis, the most popular are full-pattern fitting and the Rietveld method (Zhou et al., 2018).

2.1. XRD by Full-Pattern Fitting

Full-pattern fitting starts with rigorous sample preparation to obtain reproducible peaks. The sample must be repeatably micronized to produce reproducible particle size distributions, and spray-dried or shaken with certain solvents to prevent preferred orientation of the grains. In addition, the sample must also be spiked with a known amount of an internal standard, usually corundum or zinc oxide (Eberl, 2003). The sample is then ready for XRD analysis.

To start the analysis, the analyst chooses starting candidate phases based on their geologic knowledge of the sample and common minerals. These phases are selected from the software's library of experimentally-derived XRD patterns which were subjected to the same sample preparation described earlier. After the analyst chooses the starting phases, the software takes the selected standards and models them as a linear mixture. The software then optimizes the linear combination to find a multiplier for each standard that is related to the weight percentage of the phase in the sample.

Once the optimization chooses the best values for the multipliers, these values need to be converted to weight percentages. This is accomplished using the reference intensity ratio (RIR) method and the known amount of corundum the sample was spiked with. Each mineral phase has an RIR_{cor} value that is found by integrating the intensities of the phase's peaks across the range of angles used for analysis. This is done by summing all the intensities of the peaks and dividing that sum by the step size of the angle. The RIR_{cor} value for the corundum standard can be used to calculate the weight fraction of the mineral phase in the sample using the equation:

$$X_i = \left(\frac{X_{cor}}{RIR_{cor}}\right)\left(\frac{I_i}{I_{cor}}\right)$$

where X is the weight fraction of the mineral phase (i) and spiked corundum (cor), and I is the integrated intensity of the pattern for the mineral phase (i) and corundum (cor) (Zhou et al., 2018; Srodon, 2001).

Generally, when full-pattern fitting is referred to in this paper, it assumes the methodology found in the RockJock 11 manual (Eberl, 2003). This includes using a Cu-Kα X-ray tube, a corundum internal standard, and an analysis range of angles from 5-65° 2θ.

2.2. XRD by the Rietveld Method

2.2.1. Rietveld Analysis to Determine Phases

The Rietveld Method is another popular XRD approach. To start a Rietveld analysis, a database of reference phases and their crystallographic information is needed. Some software packages include reference phase databases, or an analyst can create their own through repositories such as the American Mineralogist Crystal Structure Database (AMCSD). Once a reference database is selected, the software identifies peaks in the sample and creates a list of possible phases from the reference database that have peaks in the same locations. The analyst then chooses which phases from the software's initial list to include and runs the quantitative phase analysis.

Rietveld analysis takes the selected phases and simulates the reflections in the sample pattern with calibrated crystallographic parameters. This is done by calculating the intensity of the peaks for each mineral phase *i* using the equation:

$$y_i(calc) = S \sum_k (p_k L_k |F_k| G(\Delta\theta_{ik}) P_k) + y_i(bkg)$$

- S = *scaling factor*
- k = k^{th} *Bragg reflection*
- p_k = *multiplicity factor*
- L_k = *Lorentz and polarization factor*
- F_k = *structure factor for individual reflection of phase*
- G = *reflection profile function*

- $\Delta\theta_{ik}$ = *Bragg angle for* k^{th} *reflection*
- P_k = *preferred orientation function*
- $y_i(bkg)$ = *refined background*

The parameters that influence the peak shapes – namely $G(\Delta\theta_{ik})$, S, P_k, and $y_i(bkg)$ – are all adjusted through an optimization procedure. Other parameters are intrinsic to the mineral phase and are therefore not adjusted. With the optimized value of S, the weight percentage of a phase in the sample can be calculated (Zhou et al., 2018). The optimized values of the other parameters can also be used to determine structural information about the mineral phases, making Rietveld a very informative method for quantitative phase analysis.

The Rietveld method is beneficial in that it is very flexible and can provide the analyst with detailed crystallographic information about the phases included in the sample (Zhou et al., 2018; Bish & Howard, 1988; Pecharsky & Zavalij, 2009). However, Rietveld analysis suffers from the same problem as full pattern fitting. The user must have experience knowing which minerals to include.

Preferred orientation (P_k parameter) is especially troublesome for Rietveld analysis because it can change peak intensities differently for each mineral phase, making some peaks shrink while others grow. If the analyst does not know how to account for preferred orientation, it can change peak intensities enough to alter which phases are suggested, and therefore included, in the analysis (Bish & Howard, 1988). Without the rigorous sample preparation described above in full-pattern fitting, it is complicated and computationally expensive to account for preferred orientation and particle size effects in Rietveld analysis. Another weakness with Rietveld analysis is applying it to non-crystalline phases (e.g., volcanic glass) or disordered phases (e.g., clay minerals) can be difficult, and in some cases impossible (Zhou et al., 2018).

2.2.2. *Calculating a Pattern from Crystal Structure*

Mineral structure information can be encoded in a uniform file format called a *.cif* file. With the crystallographic information included in a crystal structure *.cif* file, I can obtain the necessary information needed to calculate an XRD pattern (See Pecharsky & Zavalij, p. 163-176). Using the unit cell dimensions and angles, one can calculate the d-spacings for the phase (for more information on d-spacings and Miller indices, see Pecharsky & Zavalij, p. 8). The locations of the Bragg angles for each peak are directly related to the d-spacings in the crystal structure by Bragg's law:

$$sin\theta_{hkl} = \frac{\lambda}{2d_{hkl}}$$

- θ_{hkl} = *Bragg angle for Miller indices hkl*
- λ = *wavelength of X-rays used*
- d_{hkl} = *d-spacing for Miller indices hkl*

Once the angles for each peak are found, I can then calculate the intensity of the pattern at each angle *i* and apply a peak shape to each peak *k* ($1 \leq k \leq m$):

$$Y(i) = b(i) + \sum_{k=1}^{m} I_k[y_k(x_k)]$$

- *Y(i) = calculated intensity at angle i*
- b_i = *background intensity at angle i.*
- I_k = *intensity of the kth peak*
- y_k = *peak shape function*
- $x_k = 2\theta_i - 2\theta_k$

There are a few common peak shape functions used to accomplish this as detailed in Pecharsky & Zavalij (2009, p. 170-171). I utilize the Pseudo-Vogit function, which is a weighted sum of the Gaussian and Lorentzian functions:

$$y(x) = \eta \frac{C_G^{1/2}}{\sqrt{\pi H}} \exp(-C_G x^2) + (1-\eta) \frac{C_L^{\frac{1}{2}}}{\sqrt{\pi H}} (1 + C_L x^2)^{-1}$$

- *η= fractional contribution of the Gauss function*
- *$C_G = 4*\ln 2$*
- *H = full width at half maximum (FWHM)*
- *$C_L = 4$*
- *The fraction components are normalization factors for Gauss and Lorentz functions respectively such that the integrals are 1.*

2.3. Machine Learning Algorithms

Supervised machine learning algorithms are especially helpful when the data contains many weakly-predictive features (Kelleher and Tierney, 2018), such as the XRD intensity values at each angle in an XRD pattern. Therefore, the overall problem of correctly identifying the phases in an XRD pattern is a classic example of one that can be solved with supervised machine learning algorithms. I will train various machine learning algorithms with my data to determine if they are able to mimic expert analyst decisions in XRD analysis.

To determine which machine learning algorithm would be best to predict mineral presence in an XRD pattern, various machine learning algorithms were trained on an initial dataset. These were decision trees and random forests, boosted decision trees, k-nearest neighbors, support vector machines, neural networks, and logistic regression. Brief descriptions of each listed algorithm's methodology are described below.

2.3.1. Decision Trees and Random Forests

A decision tree is a classification model that creates a series of questions, the answers to which split the data into the categories of the target feature (Raschka and Mirjalili, 2017; Kelleher and Tierney, 2018). Figure 2 shows an example decision tree (Kumar, 2019). Decision trees are split into branches and nodes. Nodes (shown with boxes in the figure) represent subgroups of the dataset being classified. Branches (shown by arrows) split the data into different nodes based on values of certain features of the data. There are two main types of nodes – decision nodes and leaf nodes. Decision nodes mark the location of feature tests that split the data into different branches and subtrees. The root node is the starting decision node and contains the whole data set. Leaf nodes are ending nodes that represent homogenous subgroups of the data.

A random forest is a group of many decision trees that averages the results of each tree to determine the overall result (Raschka and Mirjalili, 2017). This method is an extension of bagged trees (Kumar, 2019). Bagging involves using bootstrapping to modify the training set for each tree and aggregates the results of each tree together to determine the final classification. This model type is less vulnerable to overfitting, more generalizable, and reduces the effect of large variance between the results of individual trees. Random forests start similarly to other bagged

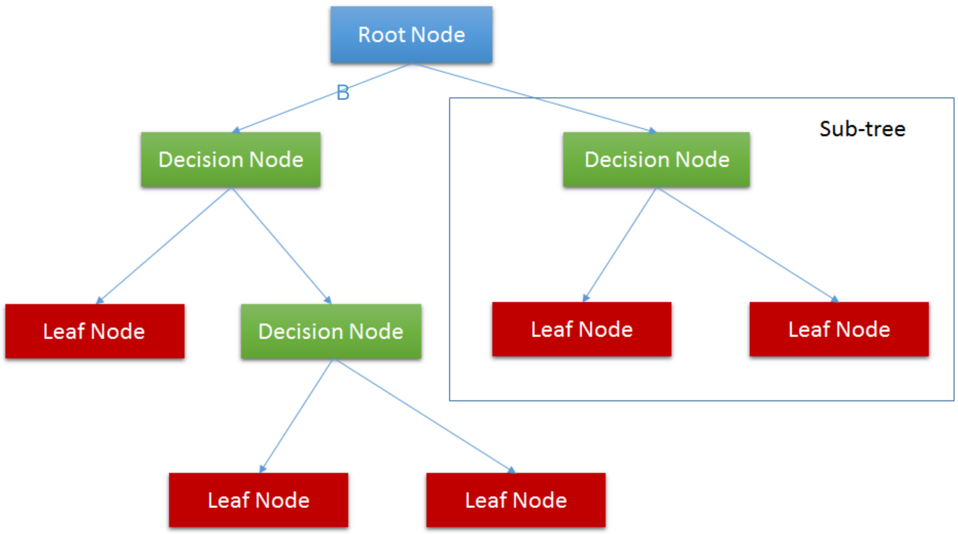

Figure 2. Decision tree terminology and basic layout.
Modified from *Machine Learning Quick Reference* (Kumar, 2019). A decision node is a location where a split is made in the data based on the values of a certain feature in the data. The root node is the starting decision node. The arrows going out of the decision nodes are branches; branches divide the tree into sub-trees and are based on a specific value of the feature being tested at the parent decision node. A leaf node is an ending node that determines a homogenous subgroup of the data.

tree models by taking a random sample of the data with replacement to use as a training set for an individual tree. Random forests differ by taking a random selection of the features without replacement at each node of the tree, usually equal to the square root of the total number of features, instead of using all features at each node. Then, the node is split according to the feature that provides the best split for the objective. This process is repeated until the desired number of trees is reached. Generally, the more trees there are in the random forest the better the model performs, with the cost of increased computation time.

2.3.2. Boosted Decision Trees

Boosting ensembles train multiple decision trees iteratively, with each iteration focusing on previously misclassified data points. Boosting is useful when decision trees and random forests contain weak learners of the objective function, in other words, they perform only slightly better than chance at correctly classifying the data (Raschka and Mirjalili, 2017). There are a few variants of boosting used to classify weak learners. Similar to random forests, one variant takes a random subset of the data without replacement to train each tree. Except for the first tree, each tree's training data is supplemented with additional data that was misclassified from the previous tree. The class chosen by the majority of the trees is used as the final prediction.

Another variation called AdaBoost trains each tree with the entire training set and adds weights to previously misclassified data (Raschka and Mirjalili, 2017). The weights are adjusted each iteration of training based on the results of the previous tree, decreasing weight for correct classifications and increasing weight for incorrect classifications. The trees are also combined with a majority vote to obtain the final classifications.

Another boosting ensemble method used in MATLAB is random undersampling (RUS) boosting. This algorithm's purpose is to classify imbalanced data, where one class contains many more samples than the other. To train the trees in the ensemble, the algorithm takes the number of samples N in the class with the fewest observations and samples N observations from every other class. Each tree in the ensemble takes a new sample of the data to use. The results of the trees are combined in a similar fashion to the other ensemble methods (MATLAB, Ensemble, 2023).

2.3.3. K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a simple algorithm that classifies the data by defining a distance metric and a number of neighbors, k (Raschka and Mirjalili, 2017). The algorithm classifies each data point by looking at the classifications of the k nearest neighbors to the point based on the distance metric. The classification of the center point is determined by the majority classification of the neighbors. If there is a tie, the algorithm chooses the classification of the neighbors that are closest to the center. KNN's main advantage is its ability to immediately adapt to new training data. However, it can consume a large amount of memory and computation time with a large data set since the algorithm requires all the training data to classify new data.

2.3.4. Support Vector Machines (SVM)

Support Vector Machines are useful to classify new data into two different groups (in this project, these groups can be a mineral of interest compared to all other minerals). The algorithm uses the mathematical concept of a kernel trick to transform the data into a higher-dimensional space, which allows for a linear classifier in a lesser dimension to classify the data (Kumar, 2019; Knox, 2018). The goal of SVM is to find the linear classifier that separates the two groups of data with the largest gap possible to avoid classification error.

2.3.5. Neural Networks

Neural networks are inspired by biological nervous systems of connected neurons (Kumar, 2019) as shown in Figure 3. Each artificial neuron is called a node, and each connection between the nodes is an edge. Starting at the input layer, each node processes the data it receives, and then sends a signal to the next layer of nodes it is connected to with an edge. Weights at each node determine the strength of that signal, and they are adjusted during the training process.

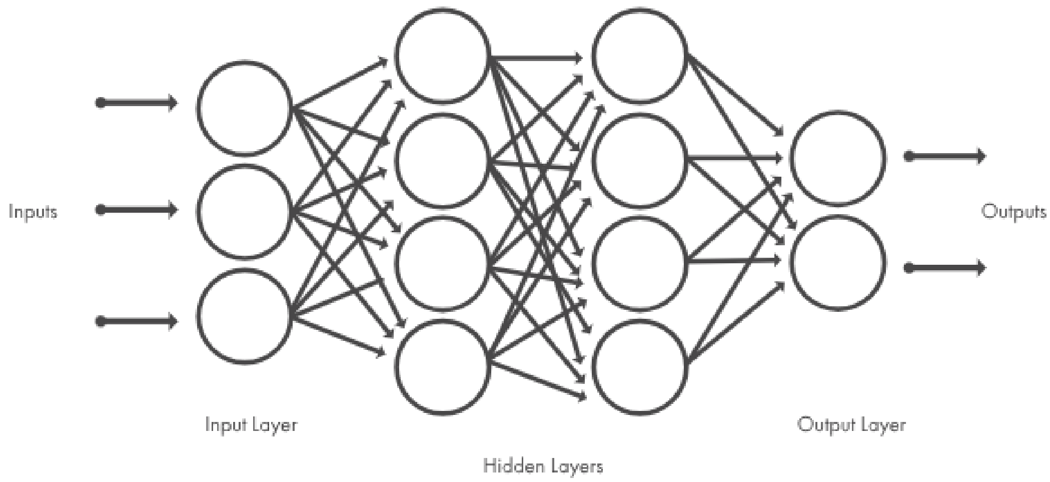

Figure 3. Neural network example.
This is an example of the structure of a neural network. Each circle represents a node, and each arrow is an edge showing the path a signal would travel through the network.
Source: "What Is a Neural Network?" https://www.mathworks.com/discovery/neural-network.html

Each layer typically performs a different transformation on the data until the signal reaches an output node and determines the classification of the sample (Kumar, 2019; Raschka & Mirjalili, 2017).

2.3.6. Logistic Regression

Logistic regression determines the probability of a class by modeling the predictor variables as a linear combination. The model optimizes the parameters for each predictor using maximum likelihood estimation iteratively until the model converges, meaning no additional improvements to the likelihood estimation are made. The optimized linear combination equals the natural log of the odds (logit function) of the event in question – in this case, the presence of the mineral phase. The use of the logit function together with the generalized linear regression model of the predictors gives it its name of logistic regression (Fenner, 2020).

3. METHODS

I tested the hypothesis that machine learning can improve the accuracy of phase-picking processes in XRD analysis. To do this, I compared the accuracy of different machine learning algorithms with baseline automation algorithms already incorporated into XRD software programs as a control. A very large data set – on the order of millions of samples – was needed to prevent overfitting in the machine learning models. Models that overfit to the training data are not able to generalize when new data is introduced and produce inaccurate results. A large dataset was also necessary to capture enough samples of rare minerals since the data generation process incorporates a mineral's abundance. Creating such a large dataset experimentally with natural or handmade mixtures would be very expensive and time consuming to produce and obtain the XRD patterns for. Therefore, I carefully created a dataset of synthetic XRD patterns and their corresponding expert knowledge analogues using data from crystal structure and mineral occurrence repositories. I trained the machine learning models with this data and evaluated their phase-picking performance with a withheld portion of the generated data as well as real mixtures, such as Reynold's Cup mixture patterns.

3.1. Synthetic Data Set

To successfully improve XRD software, I need to automate the process of narrowing down candidate phases for optimization through machine learning. To do this, a large data set of mineral mixtures with their corresponding weight percentages, XRD patterns, and elemental composition data needed to be obtained to enable the comparison of the different optimization methods without bias. I created this synthetic data set using custom MATLAB classes, crystal structure (.cif) files from the AMCSD, mineral location data from the MED, and the Rietveld equation to simulate mineral mixtures found in nature as well as information that expert XRD analysts use in their analyses.

3.1.1. Expert Knowledge Analogues

An expert analyst produces accurate XRD results by incorporating their mineralogical knowledge into their decisions of which phases to include in the optimization process. For the machine learning algorithm to create a model to mimic expert knowledge, I need data that can represent this knowledge for the model to learn from. This knowledge includes common minerals (mineral abundance), which minerals occur together (mineral co-occurrence), how XRD patterns can vary because of variations in crystal composition and structure (Jenkins, 1996), as well as results from other analysis methods, e.g., elemental composition from XRF analysis.

Mineral Abundance and Co-occurrence. To represent mineral abundance and co-occurrence, I used data from the MED. This crowd-sourced database contains nearly 300,000 reported locations of over 1.2 million mineral occurrences. The proportion of locations where a single mineral is present can represent mineral abundance. This is used during data creation as a weight when selecting from possible minerals so more abundant minerals are more likely to be chosen in a sample mixture. Mineral co-occurrence is represented in data creation by using location data from the MED to choose phases for a mixture as described in detail below in "Process for Generating Synthetic Data Samples". By using these data representations of mineral abundance and co-occurrence when creating the synthetic mixtures, I mimicked patterns that natural rock XRD patterns exhibit due to the combinations of minerals they contain. Machine learning algorithms can detect such patterns during training and use them to make more accurate predictions.

Crystal Structure Variations. Variations in crystal structure and composition are accounted for in the data set by finding a range of probable peak shape parameters for each mineral phase, as these parameters modify the peak shapes in an XRD pattern similar to how variations in a mineral influence peak shape (Pecharsky & Zavalij, 2009; Jenkins & Snyder, 1996). This is done by finding crystal structures in AMCSD that are the same phase as the standards in RockJock's library (Eberl, 2003). From these crystal structures, I calculate their XRD patterns and optimize them with the Rietveld method to match the standards' patterns. The result of the optimization is a set of realistic peak shape parameters for the crystal structure.

I optimized the peak shape parameters *U, V, W, P, X, Y*. These are related to the calculation of the width of the peak at one-half its maximum intensity, called the full width at half maximum (FWHM), which is an important measure of the peak shape. The relationship is between FWHM and the peak shape parameters (Pecharsky & Zavalij, 2009, p. 175-176) is:

$$H = (\sum_{i=0}^{5} a^i H_G^{5-i} H_L^i)^{1/5}$$

$$H_G = 2\sigma\sqrt{2\ln 2}$$

$$\sigma = \sqrt{U\,(\tan\theta)^2 + V\tan\theta + W + P/(\sin\theta)^2}$$

$$H_L = \frac{X}{\cos\theta} + \frac{Y}{\tan\theta}$$

- *a^i = tabulated coefficient that equals (1, 2.69269, 2.42843, 4.47163, 0.07842, 1) for i=0 through 5 respectively*
- *H_G = Gaussian FWHM*
- *H_L = Lorentzian FWHM*
- *U, V, W = peak broadening parameters*
- *P = additional broadening parameter*
- *X = specimen broadening parameter related to crystallite size*
- *Y = specimen broadening parameter related to microstrain*

These parameters are adequate analogues for crystal structure variations as they change the peak shapes in similar fashion to how variations such as crystallite size and microstrain affect an XRD pattern by broadening the peaks (Pecharsky & Zavalij, 2009; Jenkins & Snyder, 1996).

Initially, I hypothesized that different distributions of the peak shape parameters would be required for every mineral or mineral group (such as feldspars). However, there were not enough crystal structure files for each mineral to obtain legitimate parameter distributions. Therefore, all sets of peak shape parameters were combined into one dataset per parameter. MATLAB's Distribution Fitter App was then used to fit a distribution to each parameter. These distributions were saved as MATLAB distribution objects. While investigating the parameters, it was also discovered that the peak shape parameters in a single set are correlated with each other in some way. To preserve the relationship between parameters, I picked the parameters that had the best fitting distributions and created regression models to predict the other parameters based on a random parameter from the saved distributions. Table 1 shows the process for picking a random set of parameters.

Table 1. Random Peak Shape Parameter Generation Equations

Parameter	How Choose Parameter?	Equation/Distribution Parameters
U	Log Logistic Distribution	$\mu = -1.7947$, $\sigma = 0.6768$
V	Linear model from U	V = -0.4839 U + 0.0493
W	Linear model from U	W = 0.0484 U - 0.0049
P	Linear model from U	P = (8.659e-4) U - (5.330e-6)
X	Logistic Distribution	$\mu = 0.1626$, $\sigma = 0.0627$
Y	Linear Model from U and X	Y = 0.2133 U - 2.1149 X + 0.3867

This process gave a realistic set of peak shape parameters for a phase in the mixture samples. Since the peak shape parameters help mimic the slight variations in XRD peaks that are caused by variations in the crystal structure and so many samples were generated, a reasonable range of variations of a phase should have been accounted for in the data set. The machine learning algorithms should then be able to learn these variations for each phase and be able to correctly identify the phase despite slight variations in the XRD pattern.

Elemental Composition. Expert analysts will also use results from other analysis methods to inform their decisions on which minerals to include in their XRD analysis. XRF analysis gives insight into the elemental composition of a mixture, which can be replicated in the data. From a phase's crystal structure file, I can calculate the weight fraction of each element and represent it as a vector with one entry for each element. To determine the elemental composition of the mixture, I simply multiply each phase's elemental composition vector by the phase's corresponding weight fraction and add all scaled composition vectors together. I compared the results of machine learning model predictions with and without this chemical data included in the model to determine if it improved the accuracy of phase identification in the machine learning models.

3.1.2. Process for Generating Synthetic Data Samples

With the analogues for expert knowledge described earlier, the synthetic data samples were created. First, a location site from the MED was randomly selected. From that site, between 2-15 minerals were randomly chosen that are also included in the RockJock standards library using each mineral's abundance proportion as a weight for selection. For each mineral selected, a crystal structure for that mineral was randomly selected from the closest-matching structures in AMCSD. A corundum crystal structure was also included to represent the 20% corundum internal standard in each sample. A set of peak shape parameters was randomly chosen for each phase using the parameter distributions and regression models. Then, a realistic XRD pattern was calculated using the crystal structure information, the generated peak shape parameters, and random weight fractions for each phase by creating a SimPatternXRD object. This pattern was then distorted with normally distributed noise with a mean of -0.0118 and standard deviation of 0.1016 and normalized to the intensity of the pattern at the location of the corundum internal standard's maximum peak (43.40° 2Θ). Background intensities were not included in the simulated patterns for simplicity. This was not unreasonable as removing the background is a standard procedure before XRD analysis. Using this process, I generated 1,500,000 mixture samples to train the machine learning models.

3.2. Feature Engineering

Training machine learning algorithms with only the raw data samples can produce models that are either computationally expensive to create and use, inaccurate, or both. The solution is feature engineering – the process of extracting features (numeric representations) from aspects of the raw data and transforming them to more suitable formats for the machine learning models (Zheng & Casari, 2018). Formatting the data into meaningful features helps the machine learning algorithms create models with greater ease and produce higher quality results. Feature engineering can be as simple as representing categorical or nominal data numerically; transformations and interactions are also common types of feature engineering. For each sample, the raw features were:

- 1x3001 vector of intensities for every 0.02° 2θ of the XRD pattern (Pattern).
    - Each individual angle from 5-65° 2θ at 0.02° steps was denoted in numerical order from 1-3001. E.g., Pattern_51 would refer to the 51st angle, which is 6.00° 2θ.
- Maximum intensity of the pattern (Max_Int).

- 1x106 vector of the elemental composition weight fractions in atomic number order (Elem_Composition).
- 1x89 vector indicating which phases were present in the mixture (Minerals_Included). This was the target variable.
    - Zero indicated absence of the phase in a sample, while a one indicated the phase was present.

The engineered features for each sample were:

- 1x3000 vector for the first derivative of the XRD pattern (Pattern_1D).
    - The same naming convention used for the raw XRD pattern angles was also used for the first and second derivatives.
- 1x2999 vector for the second derivative of the XRD pattern (Pattern_2D)
- 1x9 vector of the elemental compositions for the major mineral-forming elements in atomic number order (Major_Comp).
    - Elements included were Na, Mg, Al, Si, K, Ca, Ti, Mn, and Fe.

The first and second derivatives of the pattern were chosen to give additional insight into how the pattern changes around a peak. Major elements were included in their own feature as a major element analysis is easier to obtain than a full elemental analysis.

Some groups of minerals, such as alkali and plagioclase feldspars, were expected to be difficult to identify individual phases in the machine learning models. Models for the individual phases of these mineral groups were trained, but I also created an additional response variable to represent the group. Specifically, Minerals_Included_87 corresponds to alkali feldspars, Minerals_Included_88 represents plagioclase feldspars, and Minerals_Included_89 represented all feldspars.

3.3. Feature Ranking and Selection

Feature ranking algorithms determine the most important features for distinguishing the classes so one can perform feature selection. Feature selection is a process to remove redundant and unimportant features from the model (Zheng & Casari, 2018; Ciaburro, 2017). By keeping the features that best help with prediction, feature selection helps to improve the speed of the machine learning, prevent overfitting of the models, and improve the accuracy and interpretability of the models.

3.3.1. Feature Ranking Functions

There are several functions available in MATLAB for feature ranking. These functions group the samples depending on the value of the target variable. They then perform statistical tests to determine if the values of each feature differ for each value of the target. Features that have more extreme differences in values for each target output are more important for prediction because they make it easier to distinguish which value of the target the sample should have. The functions return the list of features sorted by their importance scores for prediction (MATLAB, Feature, 2023). Based on these scores, I tested different selections of the top features to train the machine learning models.

I compared the results of three feature ranking functions on a small dataset of 1000 samples using the Classification Learner App in MATLAB using bassanite, calcite, and halite as example phases. Both XRD pattern features and elemental composition features were evaluated with the feature selection algorithms. These algorithms are Minimum Redundancy Maximum Relevance (MRMR), Analysis of Variance (ANOVA), and Kruskal Wallis.

Minimum Redundancy Maximum Relevance (MRMR). The MRMR algorithm is designed to find the features that are the most relevant to predicting the target without having mutual information with the other predictors. Mutual information between two predictors is defined as the amount one predictor’s variance can be decreased by knowing the value of the other predictor. This function assigns each feature an importance score based on how well it helps to predict the target feature while giving different information than other predictors. The features are then ordered from most important for prediction to least (MATLAB, MRMR, 2023).

Analysis of Variance (ANOVA). The ANOVA function is performed on each prediction feature. It first groups the values of the feature by the response class. It then tests the null hypothesis that the two groups of values are from populations with the same mean, versus the alternate hypothesis that the two groups come from different populations. After the test is performed for each feature, the features are ordered based on their p-values from most significant to least (MATLAB, One-way, 2023).

Kruskal Wallis. The Kruskal Wallis test is very similar to the ANOVA test, except it tests whether the groups for each feature come from populations with the same median. To compute the test statistic, it uses the rank of the data instead of their actual numeric values by ordering the data from smallest to largest across all groups and assigning an index to each instance. It is a more flexible, nonparametric variant of ANOVA that is not dependent on the data having normal distributions. Instead, it assumes the samples come from the same type of distribution (MATLAB, Kruskal, 2023).

3.4. Data Formatting

Once the features are created and selected, the data can be formatted into the different sets for the machine learning algorithms. To test the accuracy of the different models, the data needs to be split into a training set, validation set, and test set. The training set is used to initially train the various machine learning models to correctly identify the mineral phases in the samples.

These models are then tested against the validation set to see which model is the most accurate. I randomly partitioned the data into 60% training and 40% validation sets using the *cvpartition* function in MATLAB, which makes the two groups of data have roughly the same proportions of each class. The test data set consisted of 30 XRD patterns from handmade mixtures of pure minerals from past Reynold's Cup competitions, the RockJock 11 program (Eberl, 2003), and others.

A challenge with the dataset was its vast size. It was not possible to load the entire dataset into memory at once to pass it into a machine learning function. Even 10,000 samples on a personal computer would cause the code to run extremely slow or MATLAB to crash. To remedy this, I formatted the table of features to pass into the machine learning algorithms as a MATLAB tall array. This data class allows for an unlimited number of rows in the table. To make calculations on a tall array, all operations are deferred until an output is specifically requested by using the *gather* function. The tall array is then pulled into memory in smaller portions one at a time to perform the calculation. *Gather* optimizes the number of passes through the data that are needed for the calculations to obtain the requested output (MATLAB, Tall, 2023).

3.5. Apply Machine Learning Algorithms

The goal was to test if machine learning algorithms could identify phases in a mixture sample with high accuracy. Since I was only looking to identify if the minerals were included in the sample or not, a classification algorithm was the best choice. There are three main types of classification algorithms based on the classes of the target variable. The first is binary classification where there are only two mutually exclusive class labels. The second is multiclass classification. This type of model has more than two classes available, but each sample can only receive one class label. The third type is multi-label classification where each sample can have multiple class labels assigned to it.

Originally, I attempted to train a multi-label classification model that would be able to identify all the phases present in a sample. However, the Classification Learner App and the underlying classification algorithms did not support multi-label data. Even outside of the Classification Learner App, these types of models are also slow to train and complex in their methodology and interpretability. Instead, I decided to create 75 binary models – one for each target phase – which would be simpler and faster.

3.5.1. Algorithm Choice.

As described in the above in section 2.3, MATLAB has a variety of machine learning algorithms to choose from. To decide which algorithm would be the most accurate, I trained all available algorithms in MATLAB's Classification Learner App with 1000 samples for four minerals. This app allows one to train all available algorithms at once, with some variation in their hyperparameters, and visually compare their accuracy with respect to the validation set. By default, the app uses cross-validation with five folds to create the validation set. The app will split the data into five folds or sets. For each validation fold, a model is trained with the samples not in the validation fold and assessed using its performance on the validation-fold data. The average validation error is then calculated over all folds. Trained models, as well as example code to train similar models, can also be exported out of the app into MATLAB's typical workspace. The exported models are trained on the full data set which includes both the training and validation sets (MATLAB, "Select", 2023).

Table 2 shows the results of the models trained on calcite and their hyperparameter differentiations. In general, singular decision trees, random forests, boosted trees, neural networks, and some SVMs performed well with accuracies higher than 90% on the validation set (Table 2). RUS (random under-sampling) boosted trees and random forests especially did well with 98.1% and 97.3% accuracies respectively for the calcite model. RUS boosted trees looked especially promising with rare minerals like bassanite. However, the algorithm to train RUS boosted trees did not support tall arrays. In the end I chose to use the *TreeBagger* function to train random forests for each phase due to their high accuracy, tall array support, quick training times, and ease of interpretability.

Table 2. Calcite Initial Model Testing (1000 samples, All Predictors)

Model Number	Model Type	Accuracy % (Validation)	Training Time (sec)
2.25	Ensemble - RUS Boosted Trees	98.1	1887.69
2.3	Tree - Coarse	97.8	101.58
2.22	Ensemble - Bagged Trees	97.3	1652.42
2.1	Tree - Fine	96.1	121.29
2.2	Tree - Medium	96.1	105.56
2.26	Neural Network - Narrow	93.1	1799.19
2.28	Neural Network - Wide	92.5	2024.11
2.27	Neural Network - Medium	92.3	1937.65
2.30	Neural Network - Trilayered	91.9	2042.89
2.11	SVM - Cubic	91.7	413.98
2.10	SVM - Quadratic	91.6	271.13
2.29	Neural Network - Bilayered	90.6	2007.88
2.9	SVM - Linear	89.1	263.14
2.13	SVM - Gaussian	87.6	560.88
2.16	KNN - Medium	85.2	724.46
2.17	KNN - Coarse	84.9	752.98
2.20	KNN - Weighted	84.4	978.05
2.23	Ensemble - Subspace Discriminant	84.4	1571.71
2.19	KNN - Cubic	82.1	1413.19
2.14	SVM - Gaussian	79.1	559.24
2.18	KNN - Cosine	78.9	831.74
2.24	Ensemble - Subspace KNN	77.4	1880.90
2.15	KNN - Fine	75.9	624.72
2.12	SVM - Gaussian	72.8	419.98
2.21	Ensemble - AdaBoost	72.8	945.79
2.6	Logistic Regression	55	452.01

3.5.2. Balanced Datasets.

Even with the best performing algorithm, models for rare mineral phases did not perform with high sensitivity rates, in other words, they were not able to consistently identify the true positives for the phase. The models still had high accuracy simply due to the vast number of negative samples included in the data set compared to the number of positive samples. To simulate the effect that RUS boosted trees have in accounting for unbalanced classes in the data, I created smaller datasets for each phase that were balanced – they contained the same number of samples containing the phase as samples without the phase. Using these balanced datasets for each phase's model improved their sensitivity rates, dramatically decreased the training time, and simplified the workflow.

Balancing the datasets also gave an additional benefit of being able to fit the dataset into memory, eliminating the need to format the data in tall arrays once the data was balanced. This dramatically decreased the time needed to train one model from over 18 hours to about 15 minutes by obviating the need to scan through the dataset multiple times using the *gather* function. By keeping a specific phase's dataset in memory, I was also able to modify the number of samples used in hyperparameter tuning and train multiple models back-to-back in a much simpler format than was necessary with tall arrays.

It is possible that balancing the datasets took away the property of mineral abundance from the dataset. Since the data was balanced, the model may expect to see the mineral present in the sample about half the time. This could cause the models to give more false positive predictions in the test data. Low precision measures on the real data set would indicate that this may be an issue, as it measures the proportion of true positives to all predicted positives.

3.6. Algorithm Refinement and Assessment

Once all the initial models were trained with the training data sets, I refined the models by adjusting the hyperparameters. Each machine learning algorithm contains a set of hyperparameters outside of the training data for the model it creates. Examples of a hyperparameter are the number of levels created in a decision tree or the number of trees trained in a random forest. The validation set is used to determine which combination of these hyperparameters produces the best model.

The validation data was also used to determine if the models have problems with overfitting. Models that overfit to the training data perform poorly with the validation data. If overfitting is a problem with all models, it can be addressed in a few ways. One can reduce the model's complexity by either adjusting the hyperparameters – such as the number of trees in the random forest or the depth of the trees. Another solution is to simplify the data (Raschka & Mirjalili, 2017), for instance, by reducing the sample size. It is also possible to collect more training data to fix overfitting, although since I was already planning to use a vast amount of data to prevent overfitting in the first place, this was not likely to help the problem any further. Once a remedy is selected and applied, the model can be retrained and retested against the validation set until the model is no longer overfitting.

The final assessment to determine the best model for phase selection was done using the test data set of Reynold's Cup-type reference mixtures. The data were inputted to each model to determine phase classifications. Each model was assessed via 5 measures – accuracy, misclassification, sensitivity, specificity, and precision. Accuracy is the fraction of all samples that were correctly classified. Misclassification is the fraction of all samples that were misclassified. Sensitivity is the fraction of the samples containing the target phase that were correctly identified. Specificity is the fraction of the samples not containing the target phase that were classified correctly. Precision measures the fraction of the samples that were classified by the model as containing the target phase that were correct. The model that performed the best – along with the list of features used in the model – will then be made available to implement into XRD analytical software. The list of features used will help scientists prepare their data adequately for the machine learning model – in other words, if the model requires XRF chemical data to obtain the most accurate results, others can acquire that data in preparation for XRD analysis.

4. RESULTS

4.1. Feature Ranking and Selection

I tested feature ranking for three different minerals –bassanite, calcite, and halite – to gain insight into what features a machine learning model might deem important for classification, as well as the feasibility of performing feature ranking and selection for all mineral phase models.

4.1.1. Bassanite

ANOVA and Kruskal Wallis tests gave similar results in their ranking of the features. Both had the intensities around a major bassanite peak at 14.7°2θ as some of the most important features for identifying bassanite (Table 4, Table 5). However, Kruskal Wallis determined the top two most important features to be elements chromium and manganese, with chromium having a much more significant p-value of $3.8x10^{(-31)}$. The MRMR algorithm ranked elemental composition features as the most important in predicting the presence of bassanite in the samples, with titanium, chromium, and indium as the top 3 (Table 3). As bassanite does not naturally contain any of these elements, they must be associated with other minerals that occur with bassanite.

Table 3. Bassanite Feature Selection - MRMR

Feature	Score

Composition_22	1.06E-04
Composition_24	8.73E-05
Composition_49	2.07E-05
Composition_33	2.07E-05
Composition_9	2.00E-05
Composition_29	1.83E-05
Composition_25	1.79E-05
Composition_19	1.60E-05
Composition_48	1.10E-05
Composition_6	9.25E-06
Composition_30	9.05E-06
Composition_1	5.61E-06
Composition_12	5.35E-06
Composition_11	5.05E-06
Composition_16	4.73E-06
Composition_26	4.05E-06
Composition_13	3.37E-06
Composition_14	2.26E-06
Composition_20	2.05E-06
Composition_90	1.18E-06

Table 4. Bassanite Feature Selection - ANOVA

Feature	score = -log(p)	p-value
Pattern_487	218.651	2.23E-219
Pattern_486	216.302	4.99E-217
Pattern_488	210.915	1.22E-211
Pattern_485	203.503	3.14E-204
Pattern_484	197.204	6.25E-198
Pattern_483	189.674	2.12E-190
Pattern_482	187.241	5.74E-188
Pattern_481	184.232	5.87E-185
Pattern_489	180.502	3.15E-181
Pattern_479	178.357	4.40E-179
Pattern_480	166.203	6.26E-167
Pattern_478	161.307	4.93E-162
Pattern_476	157.858	1.39E-158
Pattern_477	157.632	2.33E-158
Pattern_490	132.753	1.77E-133
Pattern_475	132.323	4.75E-133
Pattern_474	121.934	1.16E-122
Pattern_473	110.632	2.33E-111
Pattern_491	95.625	2.37E-96

Pattern_472	92.587	2.59E-93

Table 5. Bassanite Feature Selection - Kruskal Wallis

Features	score = -log(p)	p-value
Composition_24	30.572	2.68E-31
Composition_25	5.932	1.17E-06
Pattern_488	5.869	1.35E-06
Pattern_487	5.869	1.35E-06
Pattern_486	5.869	1.35E-06
Pattern_489	5.862	1.37E-06
Pattern_485	5.862	1.37E-06
Pattern_484	5.856	1.39E-06
Pattern_481	5.856	1.39E-06
Pattern_476	5.856	1.39E-06
Pattern_1D_490	5.849	1.42E-06
Pattern_1D_485	5.849	1.42E-06
Pattern_490	5.849	1.42E-06
Pattern_483	5.849	1.42E-06
Pattern_482	5.849	1.42E-06
Pattern_480	5.849	1.42E-06
Pattern_479	5.849	1.42E-06
Pattern_478	5.849	1.42E-06
Pattern_477	5.849	1.42E-06
Pattern_475	5.849	1.42E-06

4.1.2. Calcite

For all three algorithms, the common important features were the intensities of the pattern around calcite's major peak at 29.40°2θ (Pattern_1221). ANOVA and Kruskal Wallis had the top important features coming only from the base XRD pattern (Table 7, Table 8), whereas the MRMR algorithm included features from the first and second derivative of the pattern around the major peak as some of the most important features for prediction (Table 6).

Table 6. Calcite Feature Selection - MRMR

Feature	Score
Pattern_1222	4.76E-01
Pattern_2D_1223	1.20E-01

Pattern_1D_1242	9.25E-02
Pattern_1D_2162	4.33E-02
Pattern_1756	4.06E-02
Pattern_2910	2.60E-02
Pattern_1D_913	2.47E-02
Pattern_2D_1229	2.44E-02
Pattern_2D_1212	2.18E-02
Pattern_1D_2194	2.13E-02
Pattern_2D_2178	1.79E-02
Pattern_2D_903	1.55E-02
Pattern_2D_2126	1.36E-02
Pattern_1D_1705	1.35E-02
Pattern_1D_2985	1.33E-02
Pattern_2D_1233	1.11E-02
Pattern_1D_2992	9.80E-03
Pattern_1D_919	7.40E-03
Pattern_1D_2791	7.00E-03
Pattern_1D_1704	6.80E-03

Table 7. Calcite Feature Selection - ANOVA

Feature	score = -log(p)	p-value
Pattern_1229	348.398	0.00E+00
Pattern_1230	345.803	0.00E+00
Pattern_1228	343.128	0.00E+00
Pattern_1227	331.444	0.00E+00
Pattern_1231	331.131	0.00E+00
Pattern_1216	325.162	0.00E+00
Pattern_1215	324.989	0.00E+00
Pattern_1217	319.278	0.00E+00
Pattern_1214	314.869	0.00E+00
Pattern_1226	314.735	0.00E+00
Pattern_1218	309.673	0.00E+00
Pattern_1232	304.478	3.33E-305
Pattern_1225	295.488	3.25E-296
Pattern_1219	295.018	9.59E-296
Pattern_1213	293.957	1.10E-294
Pattern_1220	277.711	1.94E-278
Pattern_1224	273.917	1.21E-274
Pattern_1233	265.617	2.42E-266
Pattern_1212	260.298	5.04E-261
Pattern_1221	258.564	2.73E-259

Table 8. Calcite Feature Selection - Kruskal Wallis

Features	score = -log(p)	p-value
Pattern_1222	290.064	8.64E-291
Pattern_1223	289.155	6.99E-290
Pattern_1221	289.002	9.95E-290
Pattern_1224	288.519	3.02E-289
Pattern_1220	286.745	1.80E-287
Pattern_1225	286.358	4.39E-287
Pattern_1219	283.848	1.42E-284
Pattern_1226	283.363	4.33E-284
Pattern_1218	279.550	2.82E-280
Pattern_1227	278.774	1.68E-279
Pattern_1217	273.667	2.15E-274
Pattern_1228	272.985	1.03E-273
Pattern_1216	266.556	2.78E-267
Pattern_1229	264.636	2.31E-265
Pattern_1D_1217	262.101	7.93E-263
Pattern_1D_1218	257.873	1.34E-258
Pattern_1215	257.311	4.88E-258
Pattern_1230	255.558	2.76E-256
Composition_6	253.800	1.59E-254
Pattern_1D_1219	253.497	3.18E-254

4.1.3. Halite

All three algorithms identified the presence of chlorine to be the most important feature to identify halite. After chlorine, MRMR scored the features corresponding to the first derivative of the pattern around halite's major peaks at 31.7°2θ and 45.5°2θ as most important (Table 9). ANOVA found the raw intensities of the pattern around the major peaks instead of the first derivative as the next important features after chlorine (Table 10). The top features after chlorine for Kruskal Wallis were the amount of sodium first, next the first derivative around both major peaks, and then the raw intensities around the major peaks (Table 11).

Table 9. Halite Feature Selection - MRMR

Feature	Score
Composition_17	0.0605
Pattern_1D_1349	0.0599

Pattern_1D_2023	0.0533
Pattern_1D_1334	0.0528
Pattern_1D_2037	0.052
Pattern_1D_1335	0.0471
Pattern_1D_2038	0.0453
Pattern_1D_1350	0.0449
Pattern_1D_2014	0.0394
Pattern_1D_2044	0.0332
Pattern_1D_1327	0.0321
Pattern_2026	0.0293
Pattern_1D_2565	0.0251
Pattern_1D_1361	0.0186
Pattern_1D_2045	0.0175
Pattern_1D_2022	0.0172
Pattern_1D_415	0.0134
Pattern_2D_2078	0.0134
Pattern_1D_1927	0.0134
Pattern_1D_1953	0.0134

Table 10. Halite Feature Selection - ANOVA

Feature	score = -log(p)	p-value
Composition_17	521.948	0.00E+00
Pattern_1341	335.275	0.00E+00
Pattern_1340	334.991	0.00E+00
Pattern_1339	327.497	0.00E+00
Pattern_1342	325.983	0.00E+00
Pattern_2029	315.869	0.00E+00
Pattern_2030	314.816	0.00E+00
Pattern_1338	311.830	0.00E+00
Pattern_2028	310.828	0.00E+00
Pattern_1336	309.140	0.00E+00
Pattern_1343	306.787	1.63E-307
Pattern_2031	306.122	7.55E-307
Pattern_2027	302.591	2.56E-303
Pattern_2026	300.080	8.31E-301
Pattern_1335	296.478	3.33E-297
Pattern_1337	294.981	1.05E-295
Composition_11	291.223	5.99E-292
Pattern_2024	290.274	5.33E-291
Pattern_2032	289.163	6.87E-290
Pattern_2023	281.828	1.49E-282

Table 11. Halite Feature Selection - Kruskal Wallis

Features	score = -log(p)	p-value
Composition_17	371.287	0.00E+00
Composition_11	22.269	5.38E-23
Pattern_1D_2024	18.004	9.91E-19
Pattern_1D_1336	18.004	9.91E-19
Pattern_2025	17.627	2.36E-18
Pattern_1338	17.380	4.17E-18
Pattern_2026	17.362	4.35E-18
Pattern_1337	17.362	4.35E-18
Pattern_1346	17.302	4.99E-18
Pattern_1343	17.296	5.06E-18
Pattern_1339	17.284	5.20E-18
Pattern_1340	17.254	5.57E-18
Pattern_2024	17.236	5.81E-18
Pattern_1342	17.206	6.22E-18
Pattern_1344	17.200	6.31E-18
Pattern_2028	17.194	6.40E-18
Pattern_1341	17.194	6.40E-18
Pattern_1345	17.146	7.14E-18
Pattern_1347	17.105	7.86E-18
Pattern_1348	17.039	9.14E-18

4.1.4. Feature Selection Test Models

Using three different sets of features, I tested the effect of feature selection on calcite models in the Classification Learner App with 1000 samples (Table 12). Models with a model number starting with 2 had no feature ranking algorithm applied to the predictor variables, and as such used all 9107 predictors in the model. Models starting with a 3 were trained with the top 10 predictors using the Kruskal Wallis algorithm, and models starting with a 5 were trained with the top 80 features using MRMR.

Table 12. Calcite Feature Selection Model Tests

Model Number	Model Type	Accuracy % (Validation)	Training Time (sec)	Selected Features	Feature Ranking Algorithm
2.25	Ensemble - RUS Boosted Trees	98.1	1887.69	9107/9107	None
2.3	Tree - Coarse	97.8	101.58	9107/9107	None

3.6	Logistic Regression	97.5	536.81	10 / 9107	KruskalWallis
3.30	Neural Network - Trilayered	97.4	2254.39	10 / 9107	KruskalWallis
3.20	KNN - Weighted	97.4	1382.64	10 / 9107	KruskalWallis
2.22	Ensemble - Bagged Trees	97.3	1652.42	9107/9107	None
3.22	Ensemble - Bagged Trees	97.2	1644.46	10 / 9107	KruskalWallis
3.11	SVM - Cubic	97.2	833.24	10 / 9107	KruskalWallis
3.19	KNN- Cubic	96.8	1373.51	10 / 9107	KruskalWallis
3.16	KNN - Medium	96.8	1080.10	10 / 9107	KruskalWallis
3.27	Neural Network - Medium	96.7	1969.83	10 / 9107	KruskalWallis
3.21	Ensemble - AdaBoost Trees	96.7	1633.31	10 / 9107	KruskalWallis
3.5	Discriminant - Quadratic	96.6	526.00	10 / 9107	KruskalWallis
3.25	Ensemble - RUS Boosted Trees	96.5	1925.25	10 / 9107	KruskalWallis
3.24	Ensemble - Subspace KNN	96.4	1670.95	10 / 9107	KruskalWallis
3.29	Neural Network - Bilayered	96.3	2233.59	10 / 9107	KruskalWallis
3.3	Tree - Coarse	96.2	251.94	10 / 9107	KruskalWallis
3.28	Neural Network - Wide	96.1	1975.06	10 / 9107	KruskalWallis
3.15	KNN - Fine	96.1	1067.21	10 / 9107	KruskalWallis
2.2	Tree - Medium	96.1	105.56	9107/9107	None
2.1	Tree - Fine	96.1	121.29	9107/9107	None
3.31	Kernel - SVM	96	2247.41	10 / 9107	KruskalWallis
3.26	Neural Network - Narrow	96	1952.25	10 / 9107	KruskalWallis
3.10	SVM - Quadratic	96	821.56	10 / 9107	KruskalWallis
3.9	SVM - Linear	96	815.49	10 / 9107	KruskalWallis
5.22	Ensemble - Bagged Trees	95.9	23582.73	80/9107	MRMR
3.32	Kernel - Logistic Regression	95.8	2250.71	10 / 9107	KruskalWallis
5.3	Tree - Coarse	95.8	4094.52	80/9107	MRMR
3.2	Tree - Medium	95.7	261.66	10 / 9107	KruskalWallis
5.25	Ensemble - RUS Boosted Trees	95.5	27459.98	80/9107	MRMR
3.1	Tree - Fine	95.4	251.23	10 / 9107	KruskalWallis
3.17	KNN - Coarse	95.2	1342.32	10 / 9107	KruskalWallis
3.18	KNN - Cosine	94.8	1351.71	10 / 9107	KruskalWallis
5.1	Tree - Fine	94.6	3378.21	80/9107	MRMR
5.2	Tree - Medium	94.6	3729.35	80/9107	MRMR
5.21	Ensemble - AdaBoost Trees	93.9	23290.97	80/9107	MRMR
2.26	Neural Network - Narrow	93.1	1799.19	9107/9107	None
2.28	Neural Network - Wide	92.5	2024.11	9107/9107	None
5.5	Discriminant - Quadratic	92.4	7357.22	80/9107	MRMR
2.27	Neural Network - Medium	92.3	1937.65	9107/9107	None
2.30	Neural Network - Trilayered	91.9	2042.89	9107/9107	None
2.11	SVM - Cubic	91.7	413.98	9107/9107	None
2.10	SVM - Quadratic	91.6	271.13	9107/9107	None
3.12	SVM - Fine Gaussian	91.5	849.42	10 / 9107	KruskalWallis
5.6	Logistic Regression	91.4	7830.32	80/9107	MRMR

2.29	Neural Network - Bilayered	90.6	2007.88	9107/9107	None
5.27	Neural Network - Medium	90.3	28675.50	80/9107	MRMR
5.10	SVM - Quadratic	90.2	11880.11	80/9107	MRMR
5.28	Neural Network - Wide	90.2	28749.97	80/9107	MRMR
5.9	SVM - Linear	89.5	11441.22	80/9107	MRMR
5.11	SVM - Cubic	89.4	12218.76	80/9107	MRMR
2.9	SVM - Linear	89.1	263.14	9107/9107	None
5.26	Neural Network - Narrow	88.7	27731.53	80/9107	MRMR
5.29	Neural Network - Bilayered	87.7	31500.96	80/9107	MRMR
2.13	SVM - Medium Gaussian	87.6	560.88	9107/9107	None
3.4	Discriminant - Linear	87.5	264.42	10 / 9107	KruskalWallis
3.23	Ensemble - Subspace Discriminant	87.4	1662.78	10 / 9107	KruskalWallis
5.12	SVM - Fine Gaussian	87.1	12347.01	80/9107	MRMR
5.30	Neural Network - Trilayered	86.3	31753.35	80/9107	MRMR
2.16	KNN - Medium	85.2	724.46	9107/9107	None
5.4	Discriminant - Linear	85	4152.43	80/9107	MRMR
2.17	KNN - Coarse	84.9	752.98	9107/9107	None
5.23	Ensemble - Subspace Discriminant	84.9	24520.68	80/9107	MRMR
2.23	Ensemble - Subspace Discriminant	84.4	1571.71	9107/9107	None
2.20	KNN - Weighted	84.4	978.05	9107/9107	None
5.18	KNN - Cosine	83.9	19410.37	80/9107	MRMR
2.19	KNN - Cubic	82.1	1413.19	9107/9107	None
5.20	KNN - Weighted	82.1	20474.27	80/9107	MRMR
5.31	Kernel - SVM	81.9	32360.24	80/9107	MRMR
5.16	KNN - Medium	81.1	16676.69	80/9107	MRMR
5.32	Kernel - Logistic Regression	81	32227.42	80/9107	MRMR
5.19	KNN - Cubic	80.1	20317.14	80/9107	MRMR
3.13	SVM - Medium Gaussian	79.7	1048.61	10 / 9107	KruskalWallis
5.13	SVM - Medium Gaussian	79.7	15532.30	80/9107	MRMR
2.14	SVM - Coarse Gaussian	79.1	559.24	9107/9107	None
2.18	KNN - Cosine	78.9	831.74	9107/9107	None
5.17	KNN - Coarse	78.3	19202.24	80/9107	MRMR
2.24	Ensemble - Subspace KNN	77.4	1880.90	9107/9107	None
2.15	KNN - Fine	75.9	624.72	9107/9107	None
5.24	Ensemble - Subspace KNN	75.3	24601.52	80/9107	MRMR
5.15	KNN - Fine	73.9	16430.25	80/9107	MRMR
3.14	SVM - Coarse Gaussian	72.8	1053.97	10 / 9107	KruskalWallis
2.21	Ensemble - AdaBoost Trees	72.8	945.79	9107/9107	None
2.12	SVM - Fine Gaussian	72.8	419.98	9107/9107	None
5.14	SVM - Coarse Gaussian	72.8	15837.32	80/9107	MRMR
2.60	Logistic Regression	55	452.01	9107/9107	None

Using all features still performed the best in terms of accuracy, especially when ensemble methods such as RUS boosting and bagging/random forests were used. Using fewer features interestingly improved the accuracy of other types of models that routinely had poor accuracy with all features such as logistic regression and KNN. In practice though, there was no real benefit from using fewer features in terms of accuracy.

I decided not to perform feature selection on all 86 phase models for multiple reasons. Training times of the same type of model did not improve from using fewer features. This, together with no noticeable improvement in accuracy, took away the main motivations for using feature selection. Additionally, every mineral phase would require a different set of features depending on their own XRD pattern and elemental composition, so one set of features for all models would not work unless all features were used. Finding a unique set of features to use for every phase would be costly in time and computing power. Therefore, all features were used in the models.

4.2. Machine Learning Models

This section summarizes the results of training the machine learning models and their predictions on withheld data for all types of models and on real mixture patterns for XRD only models as composition data was not available for the real data. Below are a few notes on interpreting the results in general.

When training the models, some minerals were excluded from the training for a few reasons: 1. Due to the rarity of the phase, there was not enough training data containing the phase, even in 1,500,000 samples, or it would take too long to find enough samples, and 2. None of the real data samples contained the mineral. Corundum was also excluded because there was no differentiation between corundum in the sample and the corundum added as the 20% internal standard. Table 13 shows the list of phases that had models trained and their model ID numbers.

Table 13. Mineral Phases with Trained Models

Phase Name	Model ID	RockJock ID	Phase Name	Model ID	RockJock ID
Alunite	1	1	Anorthoclase	46	91
Actinolite	2	2	Intermediate_Microcline	47	92
Tremolite	5	5	Ordered_Microcline	48	93
Analcime	6	6	Sanidine	49	95
Anatase	7	7	Magnesite	50	98
Andalusite	8	8	Magnetite	51	99
Anglesite	9	9	Marcasite	52	100
Anhydrite	10	10	Muscovite_2M1	54	103
Ankerite	11	11	Natrolite	56	106
Aragonite	12	12	Phlogopite_2M1	57	116
Arsenopyrite	13	13	Albite	58	117
Barite	14	14	Andesine	59	118
Bassanite	15	15	Anorthite	60	119
Biotite_1m	16	17	Bytownite	61	120
Calcite	17	22	Labradorite	62	121
Celestine	18	24	Oligoclase_NC	63	122
Cinnabar	21	38	Oligoclase_Norway	64	123
Cordierite	23	40	Prehnite	65	124
Diaspore	26	45	Pyrite	66	126
Dickite	27	47	Augite	67	128
Dolomite	28	48	Diopside	68	129
Dolomite_Fe_rich	29	49	Hedenbergite	69	131
Epidote	30	50	Hypersthene	70	132
Fluorapatite	31	53	Pyrrhotite	71	133
Fluorite	32	54	Quartz	72	134
Forsterite	33	55	Rutile	74	138
Galena	34	57	Sillimanite	75	142
Almandine	35	58	Silver	76	143
Grossular	36	59	Sphalerite	77	151
Gypsum	38	67	Spinel	78	152
Halite	39	68	Strontianite	79	154
Hematite	40	70	Sulfur	80	155
Illite_1M_RM30	41	75	Titanite	83	161
Illite_2M1_SG4	42	77	Tourmaline	84	163
Ilmenite	43	82	Zircon	86	168
Jarosite_Mex	44	85	Alkali Feldspars	87	-

Kaolinite_Dry_Branch	45	87	Plagioclases	88	-
			All Feldspars	89	-

The accuracy measures as defined in Section 3.6 generally range between zero and one and represent fractions of samples. Measures that have a value of -1 indicate that there were no samples of either category involved in the calculation. For example, a sensitivity of -1 indicates that the model had no predictions that were true positives or false negatives, in other words, there were no samples that had the phase included in its mixture. Instead of having a calculation of 0/0, a placeholder of -1 was used instead. In general, this indicates an absence of samples for the category. Measures equal to zero indicate that only the numerator had a zero in that measure and indicate complete misclassification in that category. All confusion matrices have the same layout, with the top left being true negatives (TN), top right is false positives (FP), bottom left is false negatives (FN), and bottom right is true positives (TP)

4.2.1. Initial Model Results

I first tested a quartz model (Table 14) from a dataset of 11,410 samples and an alunite model (Table 15) from 51253 samples as examples of a common and rare mineral, respectively. These models were trained with 40% of the data held out for a validation set. The initial performance of the models was evaluated based on prediction accuracy measures for the validation set.

Table 14. Quartz Initial Model Results

	Predicted			
	Quartz	0	1	Totals
Actual	0	2177	158	2335
	1	20	2210	2230

Accuracy	Sensitivity	Specificity	Precision
0.961	0.9252	0.991	0.9909

Table 15. Alunite Initial Model Results

		Predicted		
	Alunite	0	1	Totals
Actual	0	20352	22	20374
	1	28	87	115

Accuracy	Sensitivity	Specificity	Precision
0.9976	0.7565	0.9989	0.7982

At first glance, the alunite model seemed to perform better with an accuracy of .998 versus .961 for quartz. However, the .757 sensitivity calculation revealed that the alunite model did not perform as well for correctly identifying the samples that did contain alunite. In addition, when tested on the real data test set, the quartz model was not able to identify quartz in any sample. To attempt to improve this deficiency, I tested prediction results from other alunite models with different hyperparameters as well as bassanite and tourmaline models. The results of these model predictions on their holdout validation sets were then used to evaluate three hypotheses:

1. A balanced data set improves the accuracy of the model.
2. A model that has a sufficient number of samples of each class will give more accurate results.
3. If a pattern is more complex – i.e., contains multiple peaks that are about the same intensity – then the accuracy of the model will suffer.

In addition to these three tests, I also trained a few models of calcite with a different number of trees in the random forest to validate the choice of the default number of trees from the Classification Learner App models.

Balanced Data. Table 16 shows the confusion matrix and accuracy calculations for an alunite model trained on a balanced data set. The data contained 1300 samples with alunite and

1300 without, and 40% of the data was held out for validation. The sensitivity of the model increased to .968.

Table 16. Alunite Balanced Model

		Predicted		
	Alunite	0	1	Totals
Actual	0	518	13	531
	1	15	457	472

Accuracy	Sensitivity	Specificity	Precision
0.972	0.9682	0.9755	0.9723

The bassanite model (Table 17) was trained in the same way as the alunite model – a balanced data set with 1300 samples containing the phase and 40% holdout. This model performed extremely well by all measures, and correctly identified all samples containing bassanite. These results support the use of balanced data sets for improved prediction performance.

Table 17. Bassanite Balanced Model

		Predicted		
	Bassanite	0	1	Totals
Actual	0	521	1	522
	1	0	462	462

Accuracy	Sensitivity	Specificity	Precision
0.999	1	0.998	0.998

Sample Number. As too few and too many samples can cause overfitting to the models, I tested eleven quartz models trained on varying amounts of data to determine the optimal number of samples to use to obtain high accuracy. Since all models performed with high accuracy on the validation set, the best model was determined through the performance on a 14-sample subset of the real mixture test data (Table 18). I only used 14 of the 30 real samples because that was what

I had available at the time. The best performing model with only one misclassification was from an 1800 sample dataset (900 samples with quartz and 900 without). Therefore, all data sets were balanced to contain 1800 samples.

Table 18. Quartz Sample Size Test Model Results

Min	#Samples	Accuracy	Misclass.	Sensitivity	Specificity	Precision
72	300	0.7857	0.2143	0.75	1	1
72	400	0.8571	0.1429	0.8333	1	1
72	600	0.7143	0.2857	0.6667	1	1
72	800	0.7143	0.2857	0.6667	1	1
72	1000	0.6427	0.3571	0.5833	1	1
72	1200	0.7143	0.2857	0.6667	1	1
72	1400	0.7857	0.2143	0.75	1	1
72	1600	0.7857	0.2143	0.75	1	1
72	**1800**	**0.9286**	**0.0714**	**0.9167**	**1**	**1**
72	2000	0.7857	0.2143	0.75	1	1
72	2600	0.7857	0.2143	0.75	1	1

Pattern Complexity. To observe how the complexity of the XRD pattern might affect prediction accuracy, a tourmaline model with a balanced 2600-sample data set was tested against its 40% holdout validation data. This model had slightly lower accuracy and sensitivity measures than the other test models at 0.956 and 0.9345 respectively (Table 19). This suggests that more complex patterns might have slightly lower accuracies, but the inclusion of composition predictors might also improve that accuracy. This is further investigated in sections 4.3.3 and 4.3.4.

Table 19. Tourmaline Balanced Model

		Predicted		
	Tourmaline	0	1	Totals
Actual	0	495	11	506
	1	34	485	519

Accuracy	Sensitivity	Specificity	Precision
0.9561	0.9345	0.9783	0.9778

Number of trees. I trained three different calcite models with the same balanced data set of 1800 samples. Each model had a different number of trees in the random forest, namely 10, 30, and 100 trees (Tables 20, 21). Although the accuracy measures of the models would suggest that 10 or 100 trees would be better than 30 at first glance, the scores (the fraction of the trees in the forest that voted for each classification) for each sample are similar. The samples that were misclassified were from data that had some mistakes in the sample preparation, so the XRD pattern of the data set was not as accurate as it could have been. The scores from the other samples were very similar, mostly within 0.1 of the 30-tree sample scores. To save time on training while still achieving good accuracy measures, I decided to use 30 trees.

Table 20. Calcite Number of Trees Test Models Results

Min	#Samples	Combined Test Set				
		Accuracy	Misclass.	Sensitivity	Specificity	Precision
17	2600 30 trees	0.8571	0.1429	0.75	0.9	0.75
17	1800 30 trees	0.8571	0.1429	0.75	0.9	0.75
17	1800 10 trees	0.9286	0.0714	0.75	1	1
17	1800 100 trees	1	0	1	1	1

Table 21. Calcite Number of Trees Test Models Scores																
Min	#Samples	Calcite Correct Classifications														Class
		1	1	0	1	1	0	0	0	0	0	0	0	0	0	
		Scores														
17	2600	0.53	0.33	0.47	0.10	0.00	0.93	0.77	0.93	0.80	0.57	0.77	0.90	0.83	0.63	0
	30 trees	0.47	0.67	0.53	0.90	1.00	0.07	0.23	0.07	0.20	0.43	0.23	0.10	0.17	0.37	1
17	1800	0.40	0.73	0.50	0.30	0.07	0.93	0.80	0.83	0.83	0.90	0.77	0.80	0.77	0.77	0
	30 trees	0.60	0.27	0.50	0.70	0.93	0.07	0.20	0.17	0.17	0.10	0.23	0.20	0.23	0.23	1
17	1800	0.30	0.70	0.60	0.20	0.00	0.80	0.60	0.90	0.80	0.70	0.80	0.80	0.70	0.80	0
	10 trees	0.70	0.30	0.40	0.80	1.00	0.20	0.40	0.10	0.20	0.30	0.20	0.20	0.30	0.20	1
17	1800	0.45	0.46	0.56	0.24	0.00	0.89	0.77	0.79	0.83	0.69	0.78	0.79	0.81	0.78	0
	100 trees	0.55	0.54	0.44	0.76	1.00	0.11	0.23	0.21	0.17	0.31	0.22	0.21	0.19	0.22	1

4.2.2. All Models – XRD Only

Withheld Data Results Table 22 and Figures 4-7 show the models' results for predicting the withheld generated data. All models predicted the withheld data with accuracies above 90%, sensitivities above 86%, specificities above 88%, and precisions above 88%. Most of the models performed above 92% on all measures.

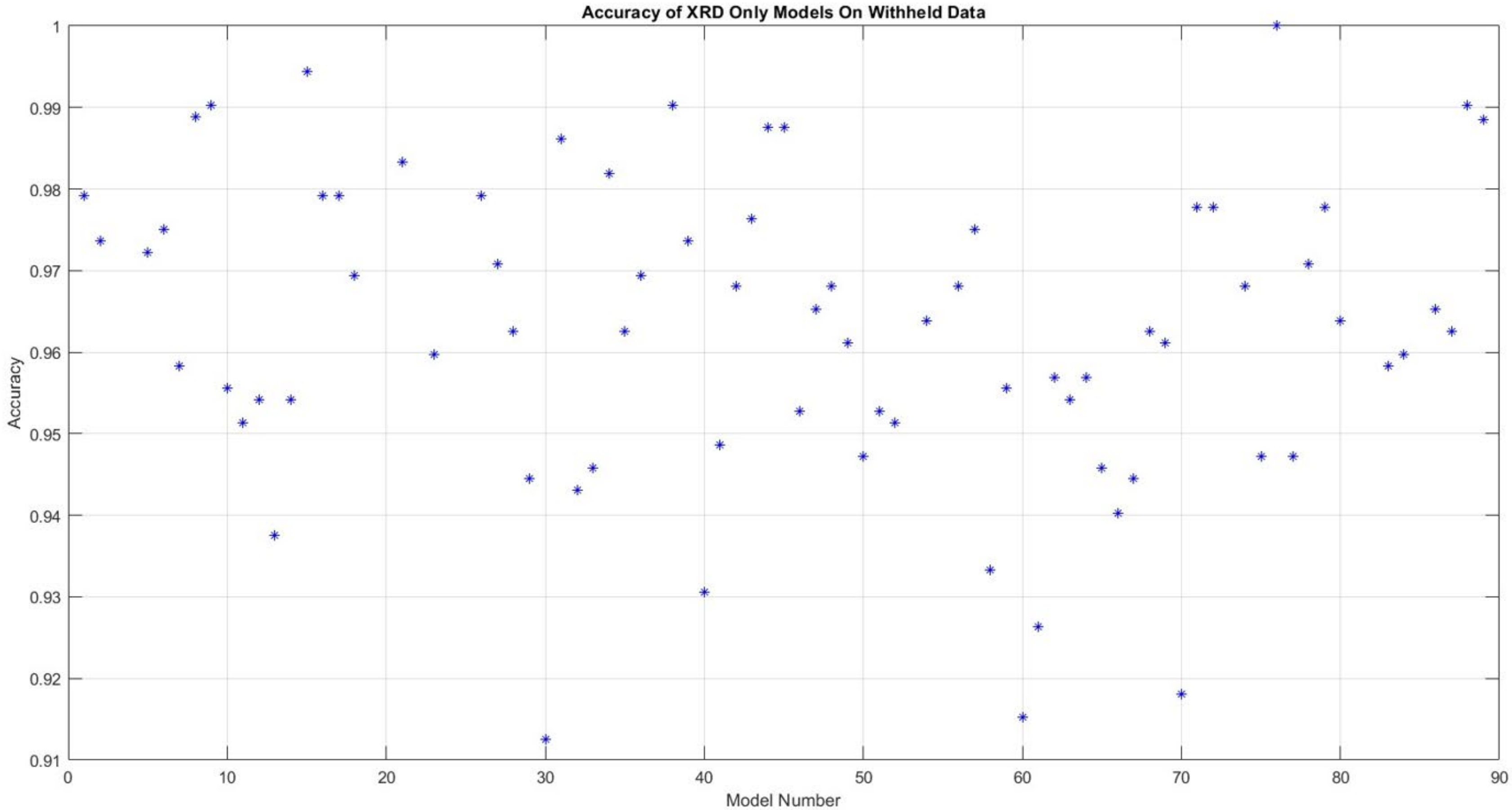

Figure 4. Accuracy of XRD-only models on withheld generated data.
All models performed with greater than 91% accuracy on the validation set of the balanced generated data. All datasets had about 720 samples, or 40%, in the validation set out of the 1800 total; the other 1080 samples were in the training set. The lowest performance minerals were epidote (30), hematite (40), albite (58), anorthite (60), bytownite (61), and hypersthene (70).

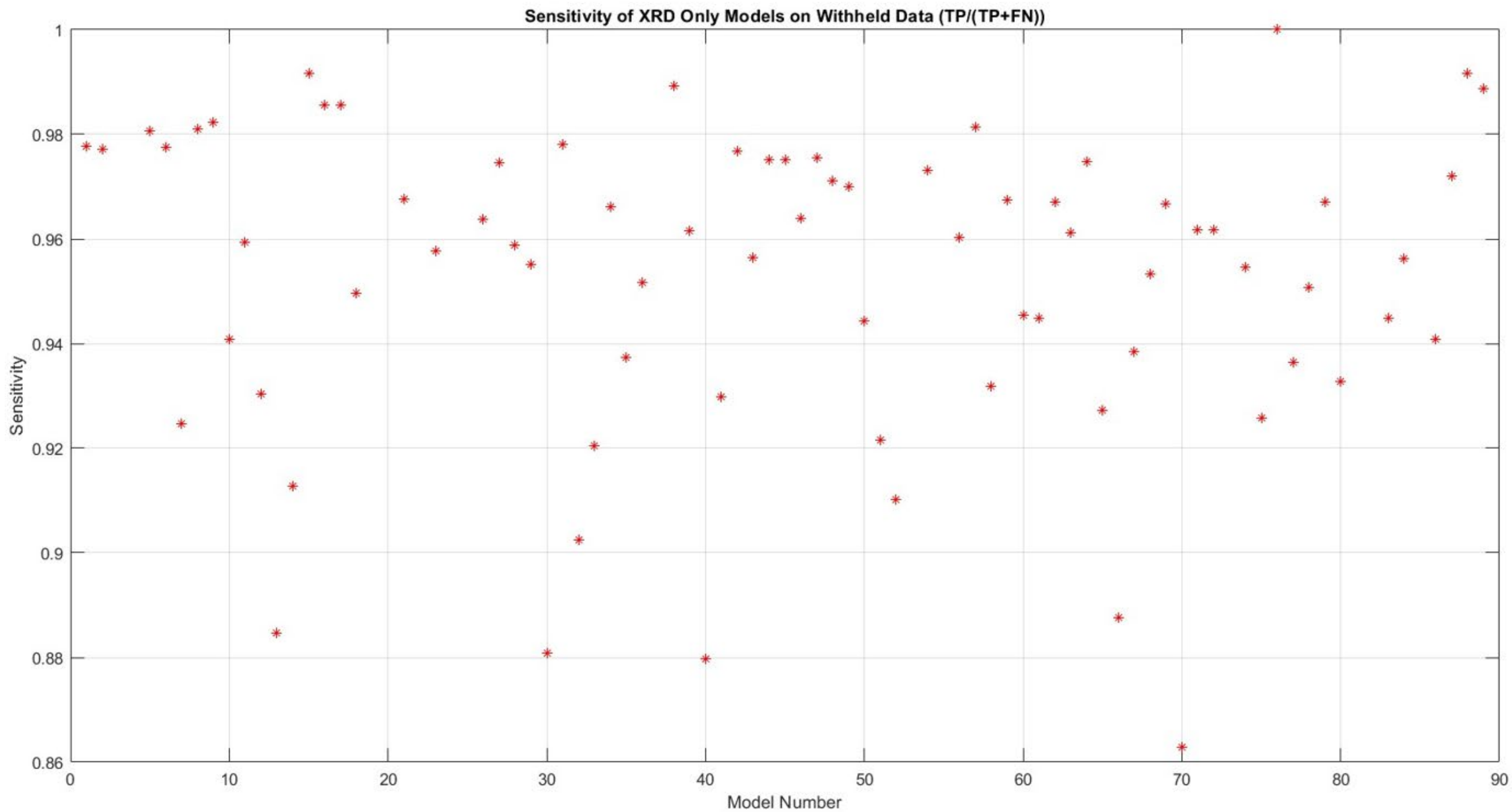

Figure 5. Sensitivity of XRD-only models on withheld generated data.
Sensitivity measures the fraction of samples containing the target phase that were correctly classified. The sensitivity for all models was above 86% on the withheld validation data. The models that did not correctly identify the phase when it was present as often were arsenopyrite (13), epidote (30), hematite (40), pyrite (66), and hypersthene (70).

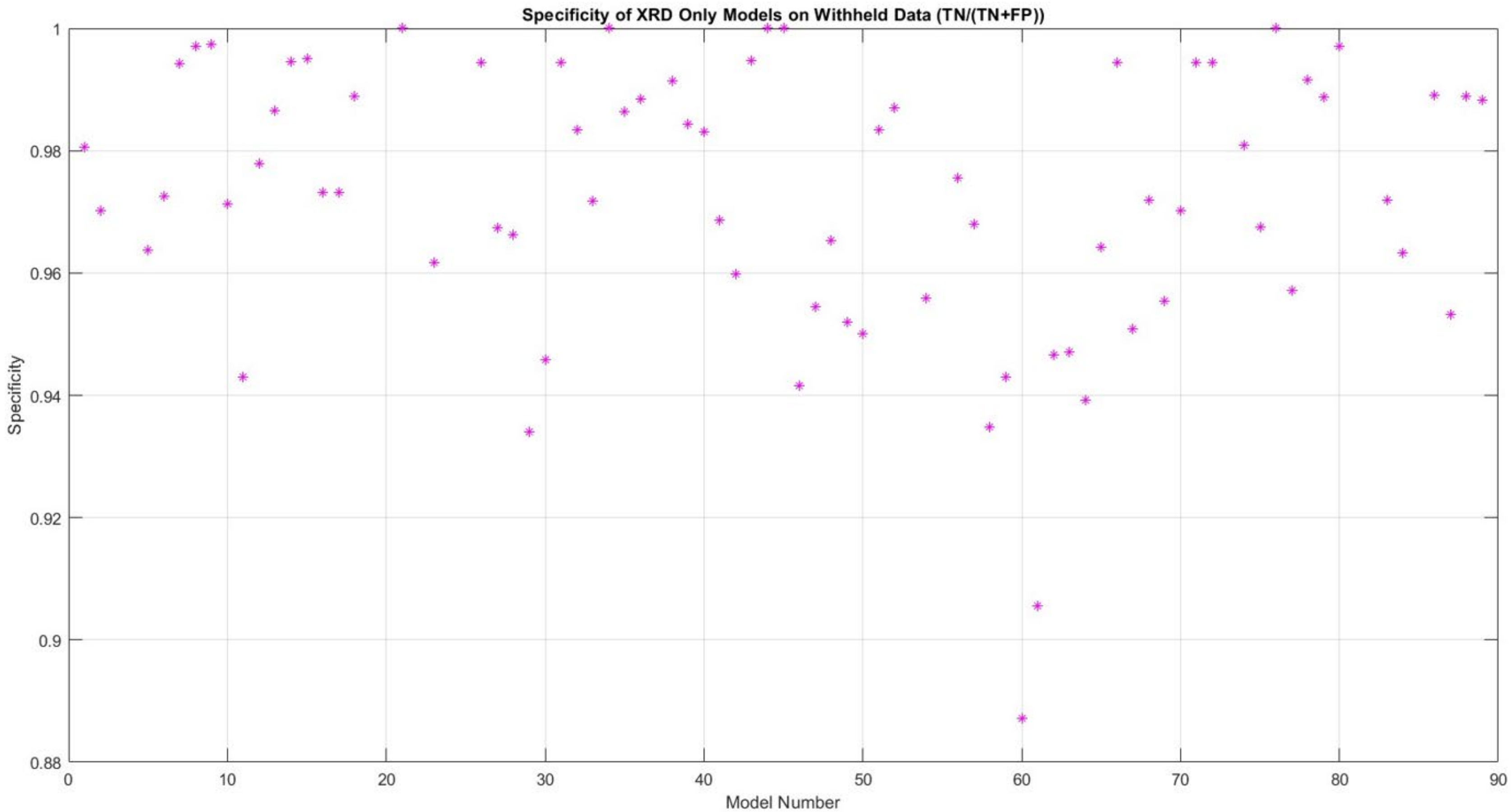

Figure 6. Specificity of XRD-only models on withheld generated data.
Specificity measures the fraction of samples not containing the target phase that were correctly classified. The specificity for all models was above 88%. The models that did not correctly identify the absence of a phase as well were anorthite (60), and bytownite (61).

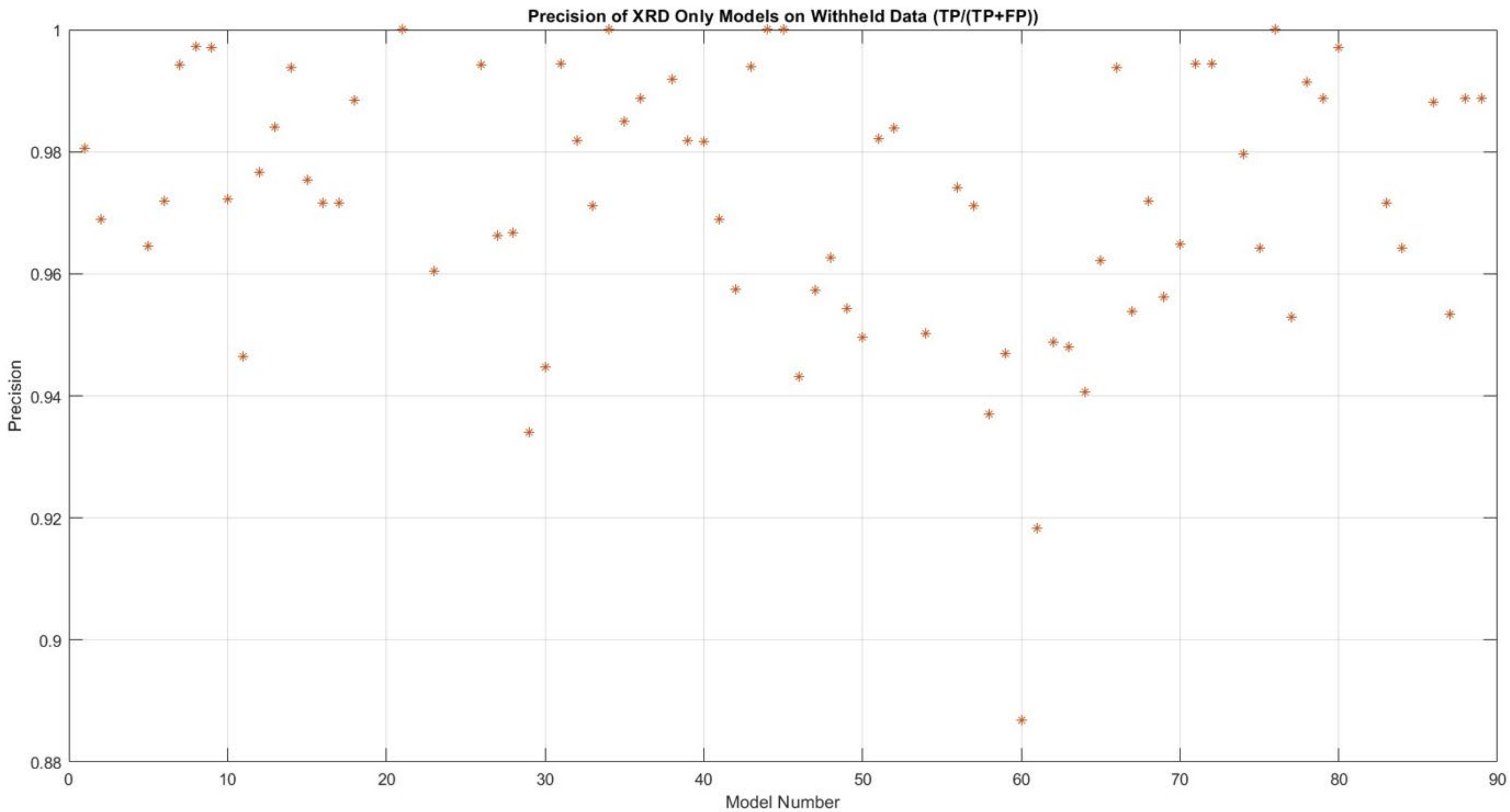

Figure 7. Precision of XRD-only models on withheld generated data.
Precision measures the fraction of samples that were predicted to contain the phase that were correct. The precision for all models was above 88%, with only anorthite (60) and bytownite (61) performing much lower than the other models.

Table 22. XRD-Only Models - Results on Validation Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Alunite	1	0.979	0.021	0.978	0.981	0.981	353	7
							8	352
Actinolite	2	0.974	0.026	0.977	0.970	0.969	358	11
							8	343
Tremolite	5	0.972	0.028	0.981	0.964	0.965	346	13
							7	354
Analcime	6	0.975	0.025	0.977	0.973	0.972	355	10
							8	347
Anatase	7	0.958	0.042	0.925	0.994	0.994	346	2
							28	344
Andalusite	8	0.989	0.011	0.981	0.997	0.997	349	1
							7	363
Anglesite	9	0.990	0.010	0.982	0.997	0.997	379	1
							6	334
Anhydrite	10	0.956	0.044	0.941	0.971	0.972	338	10
							22	350
Ankerite	11	0.951	0.049	0.959	0.943	0.947	331	20
							15	354
Aragonite	12	0.954	0.046	0.930	0.978	0.977	353	8
							25	334
Arsenopyrite	13	0.938	0.063	0.885	0.987	0.984	368	5
							40	307
Barite	14	0.954	0.046	0.913	0.995	0.994	363	2
							31	324
Bassanite	15	0.994	0.006	0.992	0.995	0.975	597	3
							1	119
Biotite_1m	16	0.979	0.021	0.986	0.973	0.972	362	10
							5	343
Calcite	17	0.979	0.021	0.986	0.973	0.972	363	4
							29	327
Celestine	18	0.969	0.031	0.950	0.989	0.988	358	4
							18	340
Cinnabar	21	0.983	0.017	0.968	1.000	1.000	349	0
							12	359
Cordierite	23	0.960	0.040	0.958	0.962	0.960	351	14
							15	340
Diaspore	26	0.979	0.021	0.964	0.994	0.994	360	2
							13	345
Dickite	27	0.971	0.029	0.975	0.967	0.966	355	12
							9	344

Table 22. XRD-Only Models - Results on Validation Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Dolomite	28	0.963	0.038	0.959	0.966	0.967	344 15	12 349
Dolomite_ Fe_rich	29	0.944	0.056	0.955	0.934	0.934	340 16	24 340
Epidote	30	0.913	0.088	0.881	0.946	0.945	332 44	19 325
Fluorapatite	31	0.986	0.014	0.978	0.994	0.994	353 8	2 357
Fluorite	32	0.943	0.057	0.903	0.983	0.982	355 35	6 324
Forsterite	33	0.946	0.054	0.921	0.972	0.971	345 29	10 336
Galena	34	0.982	0.018	0.966	1.000	1.000	335 13	0 372
Almandine	35	0.963	0.038	0.937	0.986	0.985	364 22	5 329
Grossular	36	0.969	0.031	0.952	0.989	0.989	344 18	4 354
Gypsum	38	0.990	0.010	0.989	0.991	0.992	346 4	3 367
Halite	39	0.974	0.026	0.962	0.984	0.982	376 13	6 325
Hematite	40	0.931	0.069	0.880	0.983	0.982	348 44	6 322
Illite_1M_ RM30	41	0.949	0.051	0.930	0.969	0.969	339 26	11 344
Illite_2M1_SG4	42	0.968	0.032	0.977	0.960	0.958	359 8	15 338
Ilmenite	43	0.976	0.024	0.956	0.995	0.994	374 15	2 329
Jarosite_Mex	44	0.988	0.013	0.975	1.000	1.000	359 9	0 352
Kaolinite_ Dry_Branch	45	0.988	0.013	0.975	1.000	1.000	353 10	1 347
Anorthoclase	46	0.953	0.047	0.964	0.942	0.943	338 13	21 348
Intermediate_ Microcline	47	0.965	0.035	0.976	0.955	0.957	336 9	16 359

Table 22. XRD-Only Models - Results on Validation Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Ordered_ Microcline	48	0.968	0.032	0.971	0.965	0.963	362 10	13 335
Sanidine	49	0.961	0.039	0.970	0.952	0.954	337 11	17 355
Magnesite	50	0.947	0.053	0.944	0.950	0.950	343 20	18 339
Magnetite	51	0.953	0.047	0.922	0.983	0.982	357 28	6 329
Marcasite	52	0.951	0.049	0.910	0.987	0.984	381 30	5 304
Muscovite_ 2M1	54	0.964	0.036	0.973	0.956	0.950	369 9	17 325
Natrolite	56	0.968	0.032	0.960	0.976	0.974	359 14	9 338
Phlogopite_ 2M1	57	0.975	0.025	0.981	0.968	0.971	332 7	11 370
Albite	58	0.933	0.067	0.932	0.935	0.937	330 25	23 342
Andesine	59	0.956	0.044	0.967	0.943	0.947	331 12	20 357
Anorthite	60	0.915	0.085	0.945	0.887	0.887	330 19	42 329
Bytownite	61	0.926	0.074	0.945	0.906	0.918	307 21	32 360
Labradorite	62	0.957	0.043	0.967	0.947	0.949	337 12	19 352
Oligoclase_NC	63	0.954	0.046	0.961	0.947	0.948	340 14	19 347
Oligoclase-Norway	64	0.957	0.043	0.975	0.939	0.941	340 9	22 349
Prehnite	65	0.946	0.054	0.927	0.964	0.962	350 26	13 331
Pyrite	66	0.940	0.060	0.888	0.994	0.994	353 41	2 324
Augite	67	0.944	0.056	0.939	0.951	0.954	329 23	17 351
Diopside	68	0.963	0.038	0.953	0.972	0.972	346 17	10 347

Table 22. XRD-Only Models - Results on Validation Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Hedenbergite	69	0.961	0.039	0.967	0.955	0.956	343	16
							12	349
Hypersthene	70	0.918	0.082	0.863	0.970	0.965	359	11
							48	302
Pyrrhotite	71	0.978	0.022	0.962	0.994	0.994	352	2
							14	352
Quartz	72	0.978	0.022	0.962	0.994	0.994	358	2
							33	328
Rutile	74	0.968	0.032	0.955	0.981	0.980	360	7
							16	337
Sillimanite	75	0.947	0.053	0.926	0.968	0.964	358	12
							26	324
Silver	76	1.000	0.000	1.000	1.000	1.000	347	0
							0	373
Sphalerite	77	0.947	0.053	0.936	0.957	0.953	358	16
							22	324
Spinel	78	0.971	0.029	0.951	0.992	0.991	352	3
							18	347
Strontianite	79	0.978	0.022	0.967	0.989	0.989	351	4
							12	353
Sulfur	80	0.964	0.036	0.933	0.997	0.997	347	1
							25	347
Titanite	83	0.958	0.042	0.945	0.972	0.972	347	10
							20	343
Tourmaline	84	0.960	0.040	0.956	0.963	0.964	341	13
							16	350
Zircon	86	0.965	0.035	0.941	0.989	0.988	361	4
							21	334
K-feldspars	87	0.963	0.038	0.972	0.953	0.953	346	17
							10	347
Plagioclases	88	0.990	0.010	0.992	0.989	0.989	359	4
							3	354
All Feldspars	89	0.988	0.012	0.989	0.988	0.989	503	6
							6	525

Real Data Results Table 23 and Figures 8-11 show the accuracy measures for each model's predictions on the real data. Most models performed with greater than 90% accuracy on the 30 real samples (Figure 8). Phases that were frequently misidentified in the real data samples with less than 60% accuracy included dickite; dolomite; intermediate microcline, ordered microcline, and sanidine; albite; bytownite; labradorite; oligoclase; and diopside. From the table of results in Table 23 I can see that these models classified too many samples as having the mineral present. This could be due to the balanced data sets, which removed any dependence on mineral abundance from the training sets. This is a limitation that could be resolved in the future by generating more data so enough samples containing the mineral could be present while still preserving the abundance ratio.

Feldspars was predicted with very low accuracies on the real data, even after combining each feldspar group into one model. The only models that had higher than 60% accuracy were specific phases that were not included in the majority of the real samples. The phases that were included in the real samples had models that were not able to identify the phase correctly in almost all the samples.

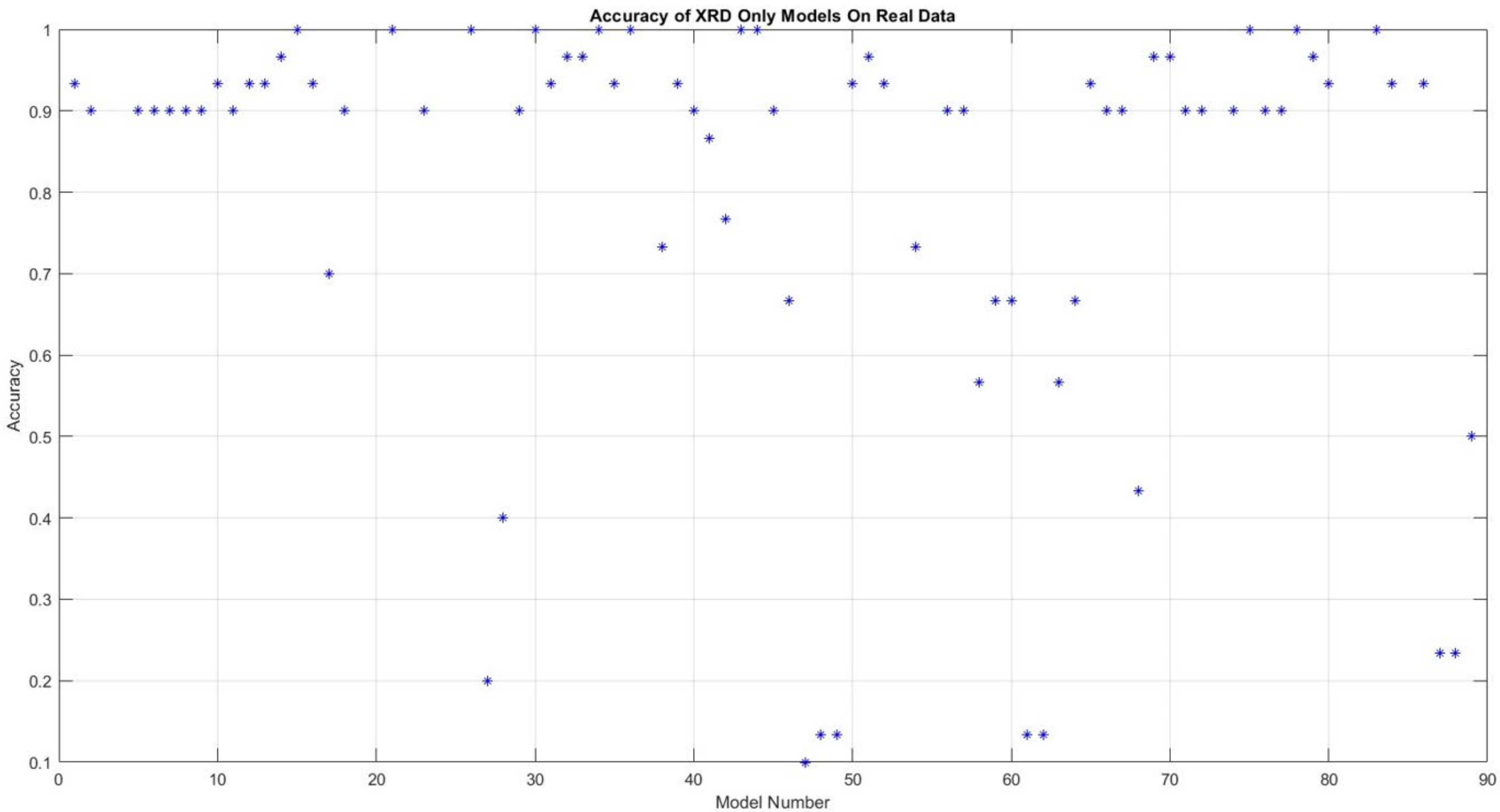

Figure 8. Accuracy of XRD-only models on real data.
Accuracy varied dramatically when tested on physical mixtures' XRD patterns. 80% of the models performed above 80% accuracy. Out of the 20 models that performed below 80%, 8 of them were above 60%. The models that performed worse than chance were dickite (27), dolomite (28), intermediate microcline (47), ordered microcline (48), sanidine (49), bytownite (61), labradorite (62), diopside (68), alkali feldspars (87), and plagioclases (88).

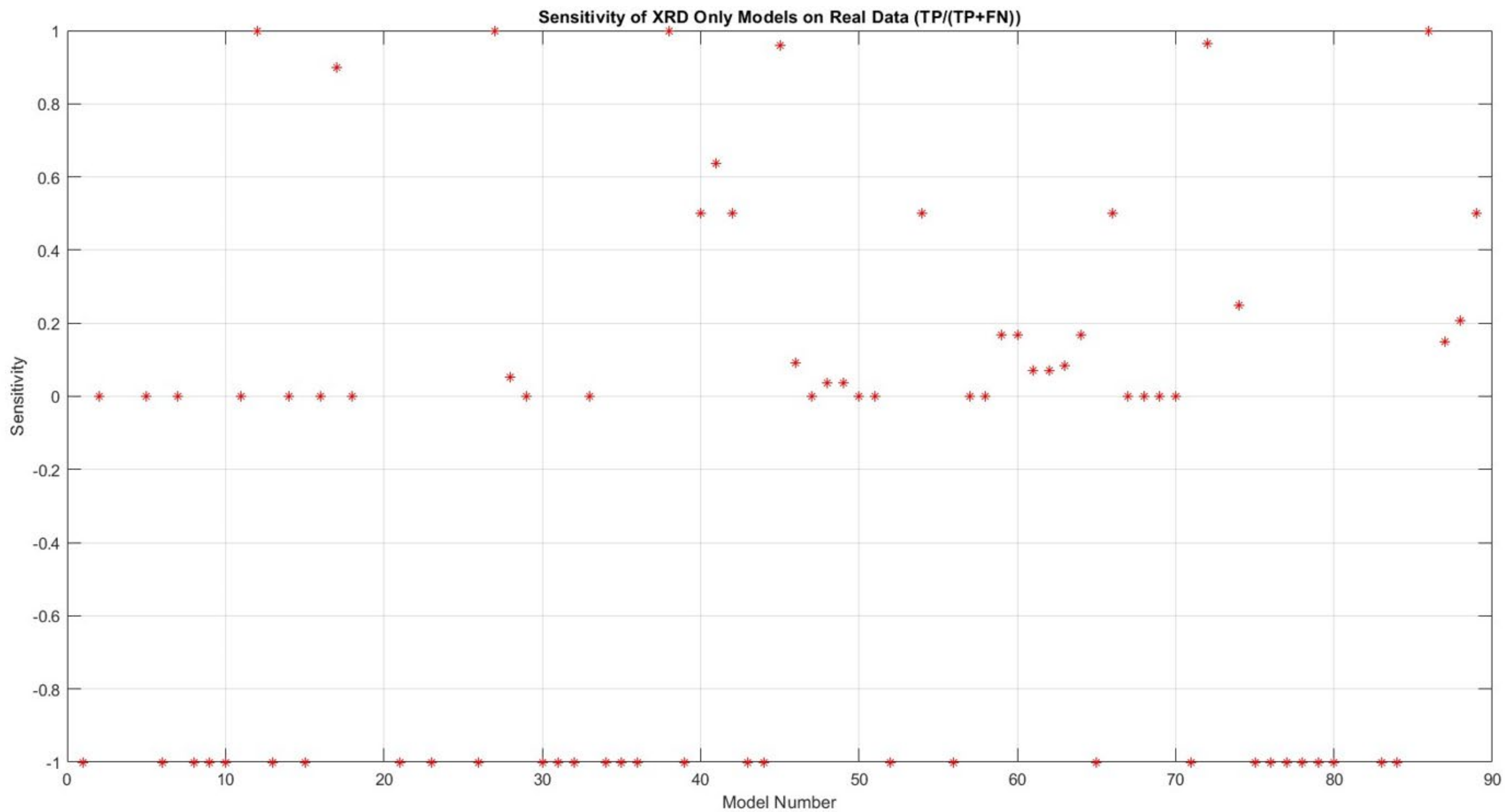

Figure 9. Sensitivity of XRD-only models on real data.
Many models had a sensitivity of 0 on the real mixtures, meaning the model incorrectly classified all the samples containing the target phase. The models with samples of -1 correspond to phases that were not present in any of the real mixtures. This can correspond to high accuracy if there were low numbers of false positives as well.

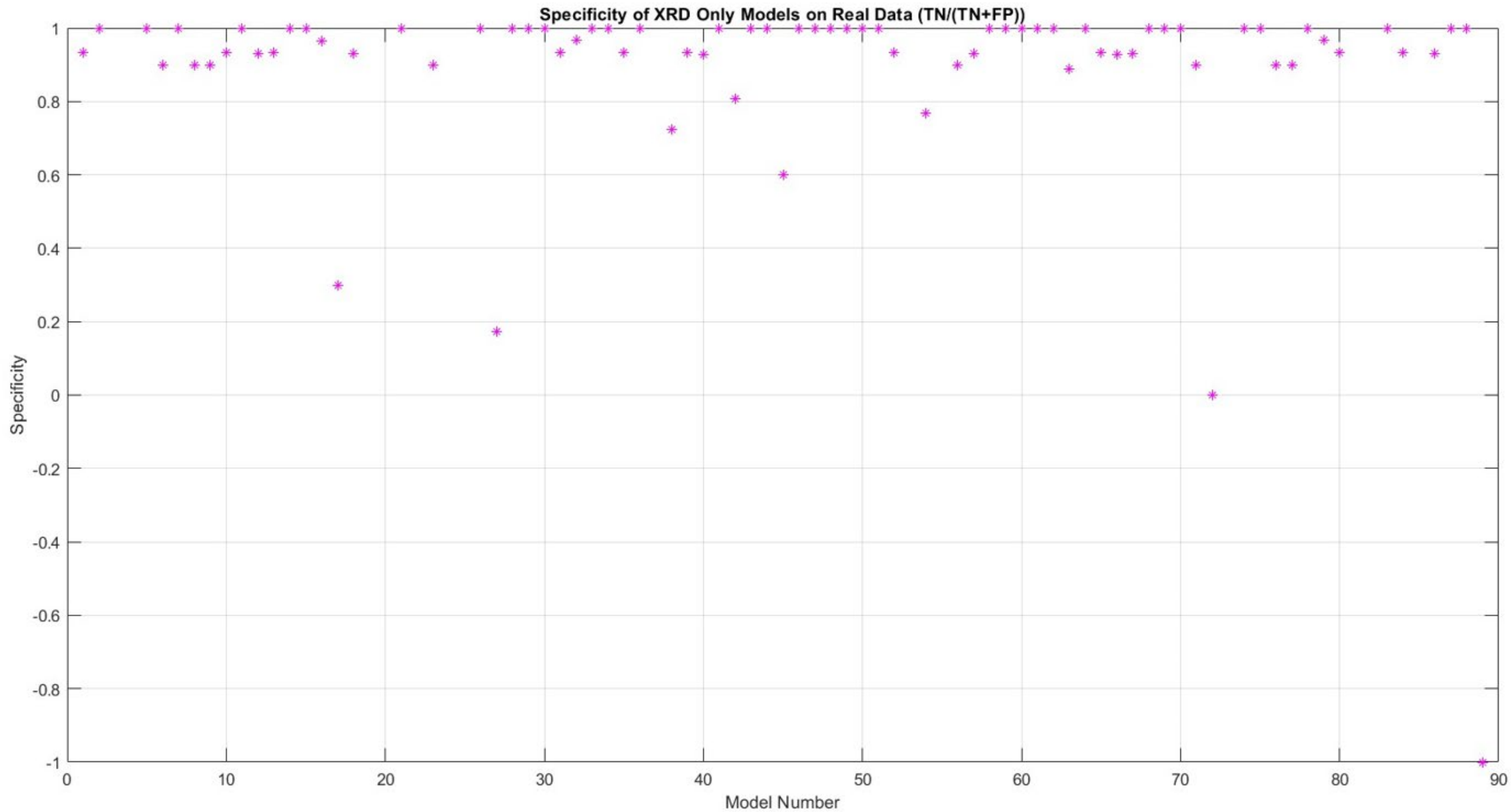

Figure 10. Specificity of XRD-only models on real data.
Most models performed well on identifying samples where the target phase was absent. The only model with a specificity of -1 was the all-feldspars model because all samples had at least one type of feldspar included. Models that usually classified samples as containing the phase when they did not were calcite (17), dickite (27), kaolinite (45), and quartz (72). Many models, however, were able to perfectly identify the samples where the target phase was absent.

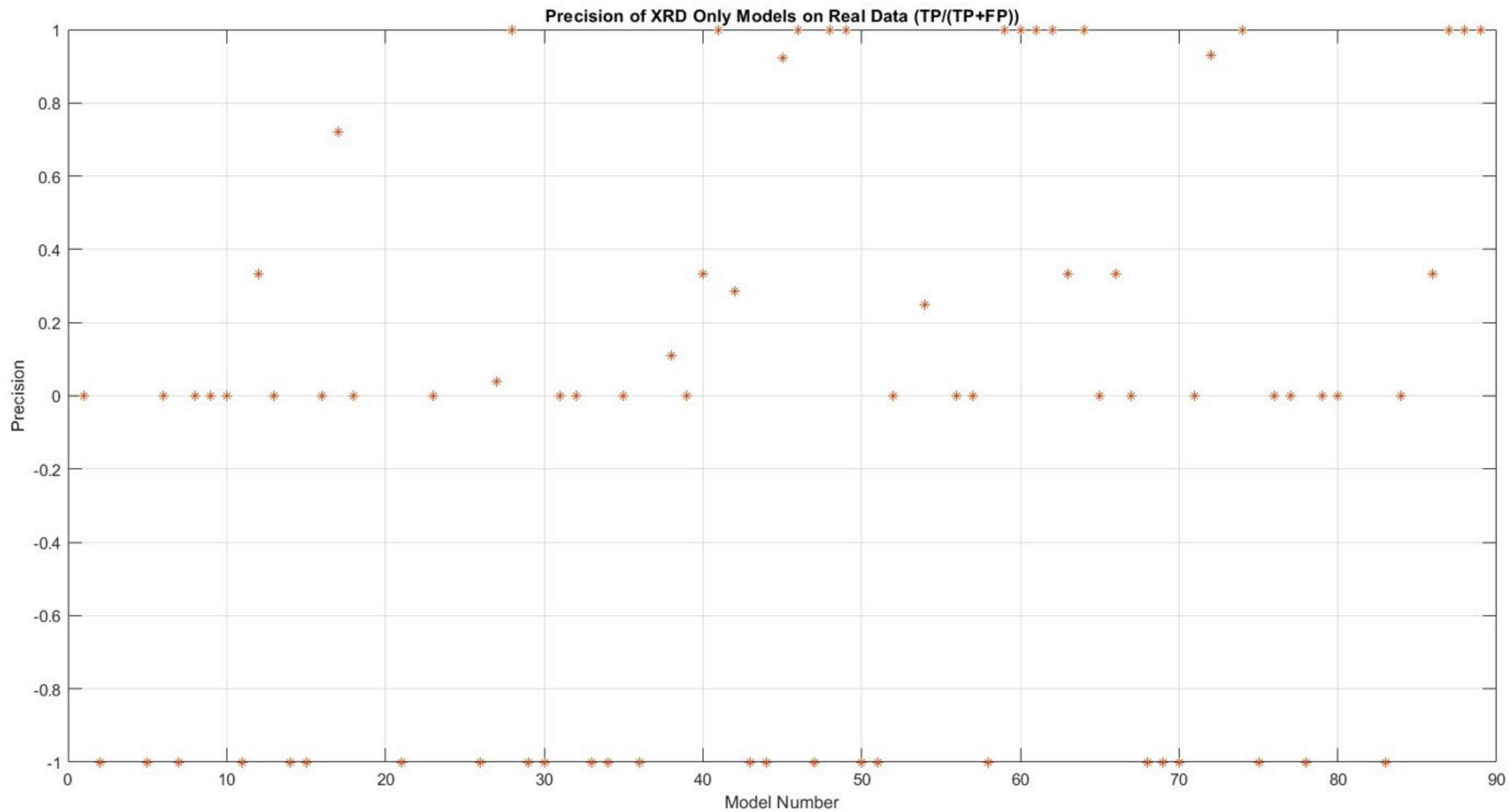

Figure 11. Precision of XRD-only models on real data.
Many models had values of -1 for precision, meaning there were no samples that the model classified as containing the target phase. This implies a few things: 1. All the negative samples were correctly classified and correspond to a specificity=1 and 2. If there were any samples containing the target, they were all misclassified and sensitivity would equal 0. If there were no samples containing the target, accuracy=1. Many models also had zero precision, meaning all the samples that were classified as containing the phase were false positives.

Table 23. XRD-Only Models – Results on Real Test Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Alunite	1	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Actinolite	2	0.9000	0.1000	0.0000	1.0000	-1.0000	27	0
							3	0
Tremolite	5	0.9000	0.1000	0.0000	1.0000	-1.0000	27	0
							3	0
Analcime	6	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Anatase	7	0.9000	0.1000	0.0000	1.0000	-1.0000	27	0
							3	0
Andalusite	8	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Anglesite	9	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Anhydrite	10	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Ankerite	11	0.9000	0.1000	0.0000	1.0000	-1.0000	27	0
							3	0
Aragonite	12	0.9333	0.0667	1.0000	0.9310	0.3333	27	2
							0	1
Arsenopyrite	13	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Barite	14	0.9667	0.0333	0.0000	1.0000	-1.0000	29	0
							1	0
Bassanite	15	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Biotite_1m	16	0.9333	0.0667	0.0000	0.9655	0.0000	28	1
							1	0
Calcite	17	0.7000	0.3000	0.9000	0.3000	0.7200	3	7
							2	18
Celestine	18	0.9000	0.1000	0.0000	0.9310	0.0000	27	2
							1	0
Cinnabar	21	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Cordierite	23	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Diaspore	26	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0

Table 23. XRD-Only Models – Results on Real Test Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Dickite	27	0.2000	0.8000	1.0000	0.1724	0.0400	5	24
							0	1
Dolomite	28	0.4000	0.6000	0.0526	1.0000	1.0000	11	0
							18	1
Dolomite_Fe_rich	29	0.9000	0.1000	0.0000	1.0000	-1.0000	27	0
							3	0
Epidote	30	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Fluorapatite	31	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Fluorite	32	0.9667	0.0333	-1.0000	0.9667	0.0000	29	1
							0	0
Forsterite	33	0.9667	0.0333	0.0000	1.0000	-1.0000	29	0
							1	0
Galena	34	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Almandine	35	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Grossular	36	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Gypsum	38	0.7333	0.2667	1.0000	0.7241	0.1111	21	8
							0	1
Halite	39	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Hematite	40	0.9000	0.1000	0.5000	0.9286	0.3333	26	2
							1	1
Illite_1M_RM30	41	0.8667	0.1333	0.6364	1.0000	1.0000	19	0
							4	7
Illite_2M1_SG4	42	0.7667	0.2333	0.5000	0.8077	0.2857	21	5
							2	2
Ilmenite	43	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Jarosite_Mex	44	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Kaolinite_ Dry_Branch	45	0.9000	0.1000	0.9600	0.6000	0.9231	3	2
							1	24
Anorthoclase	46	0.6667	0.3333	0.0909	1.0000	1.0000	19	0
							10	1

Table 23. XRD-Only Models – Results on Real Test Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Intermediate_ Microcline	47	0.1000	0.9000	0.0000	1.0000	-1.0000	3 27	0 0
Ordered_ Microcline	48	0.1333	0.8667	0.0370	1.0000	1.0000	3 26	0 1
Sanidine	49	0.1333	0.8667	0.0370	1.0000	1.0000	3 26	0 1
Magnesite	50	0.9333	0.0667	0.0000	1.0000	-1.0000	28 2	0 0
Magnetite	51	0.9667	0.0333	0.0000	1.0000	-1.0000	29 1	0 0
Marcasite	52	0.9333	0.0667	-1.0000	0.9333	0.0000	28 0	2 0
Muscovite_2M1	54	0.7333	0.2667	0.5000	0.7692	0.2500	20 2	6 2
Natrolite	56	0.9000	0.1000	-1.0000	0.9000	0.0000	27 0	3 0
Phlogopite_2M1	57	0.9000	0.1000	0.0000	0.9310	0.0000	27 1	2 0
Albite	58	0.5667	0.4333	0.0000	1.0000	-1.0000	17 13	0 0
Andesine	59	0.6667	0.3333	0.1667	1.0000	1.0000	18 10	0 2
Anorthite	60	0.6667	0.3333	0.1667	1.0000	1.0000	18 10	0 2
Bytownite	61	0.1333	0.8667	0.0714	1.0000	1.0000	2 26	0 2
Labradorite	62	0.1333	0.8667	0.0714	1.0000	1.0000	2 26	0 2
Oligoclase_NC	63	0.5667	0.4333	0.0833	0.8889	0.3333	16 11	2 1
Oligoclase_Norway	64	0.6667	0.3333	0.1667	1.0000	1.0000	18 10	0 2
Prehnite	65	0.9333	0.0667	-1.0000	0.9333	0.0000	28 0	2 0
Pyrite	66	0.9000	0.1000	0.5000	0.9286	0.3333	26 1	2 1
Augite	67	0.9000	0.1000	0.0000	0.9310	0.0000	27 1	2 0

Table 23. XRD-Only Models – Results on Real Test Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Diopside	68	0.4333	0.5667	0.0000	1.0000	-1.0000	13	0
							17	0
Hedenbergite	69	0.9667	0.0333	0.0000	1.0000	-1.0000	29	0
							1	0
Hypersthene	70	0.9667	0.0333	0.0000	1.0000	-1.0000	29	0
							1	0
Pyrrhotite	71	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Quartz	72	0.9000	0.1000	0.9643	0.0000	0.9310	0	2
							1	27
Rutile	74	0.9000	0.1000	0.2500	1.0000	1.0000	26	0
							3	1
Sillimanite	75	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Silver	76	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Sphalerite	77	0.9000	0.1000	-1.0000	0.9000	0.0000	27	3
							0	0
Spinel	78	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Strontianite	79	0.9667	0.0333	-1.0000	0.9667	0.0000	29	1
							0	0
Sulfur	80	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Titanite	83	1.0000	0.0000	-1.0000	1.0000	-1.0000	30	0
							0	0
Tourmaline	84	0.9333	0.0667	-1.0000	0.9333	0.0000	28	2
							0	0
Zircon	86	0.9333	0.0667	1.0000	0.9310	0.3333	27	2
							0	1
K-feldspars	87	0.2333	0.7667	0.1481	1.0000	1.0000	3	0
							23	4
Plagioclases	88	0.2333	0.7667	0.2069	1.0000	1.0000	1	0
							23	6
All Feldspars	89	0.5000	0.5000	0.5000	-1.0000	1.0000	0	0
							15	15

4.2.3. All Models – XRD + Major Elements

As there were not any composition data available for the real mixtures, the XRD + major elements models were only tested on the validation portion of the data sets. All models predicted the validation set with greater than 91% accuracy, 86% sensitivity, 91% specificity, and 91% precision (Table 24, Figures 12-15). This implies that all models were excellent at identifying when a phase was not present in the sample but were not quite as reliable when the phase was present. Various feldspars were the only models that had an accuracy of lower than 94%. There is no visible improvement of the model results by adding in the major elements compared to just the XRD pattern data.

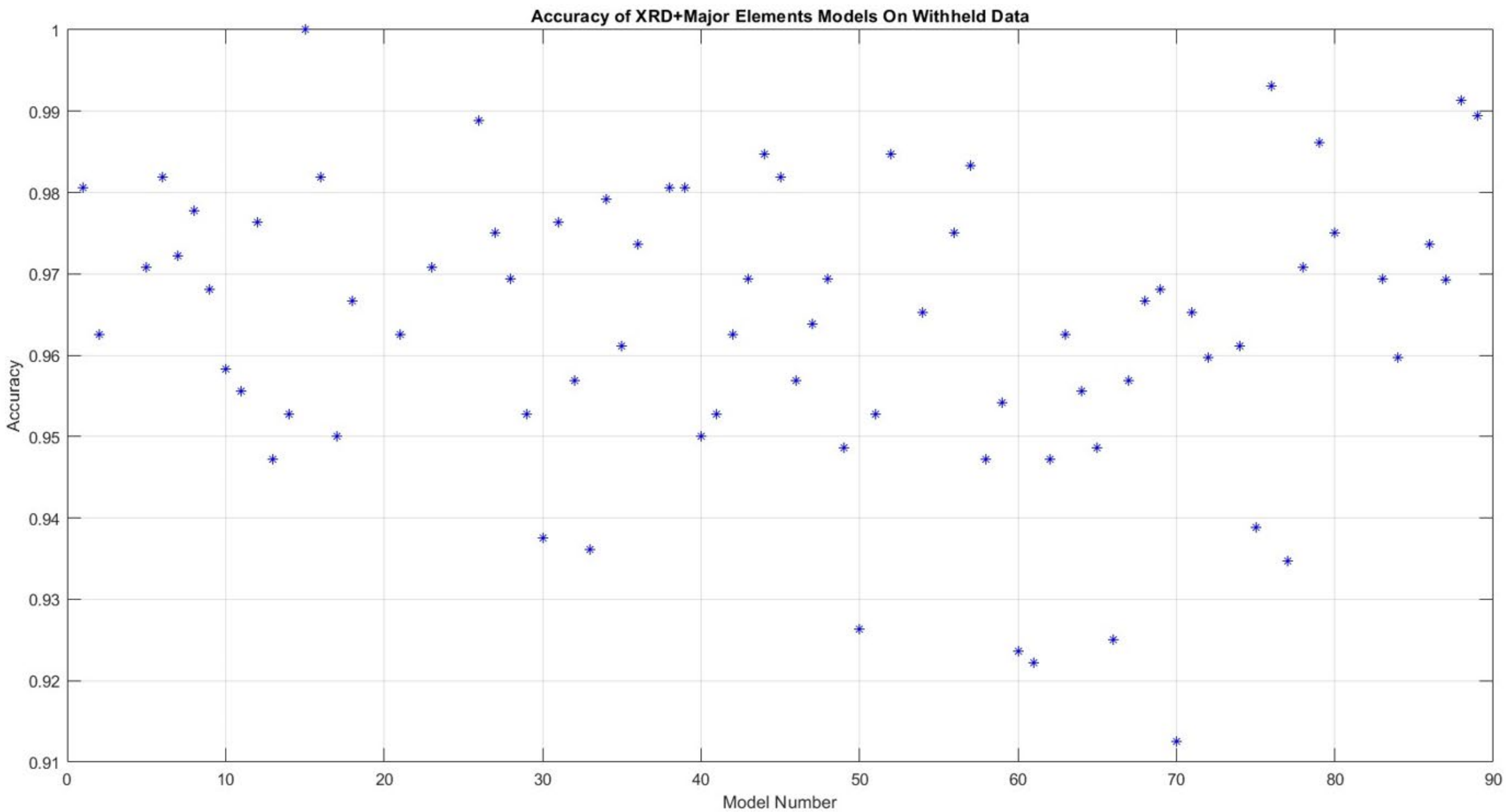

Figure 12. Accuracy of XRD + Major Elements models on withheld generated data.
The majority of the models performed with higher than 95% accuracy in their predictions on the withheld data. The lowest performing models were magnesite (50), anorthite (60), bytownite (61), pyrite (66), and hypersthene (70).

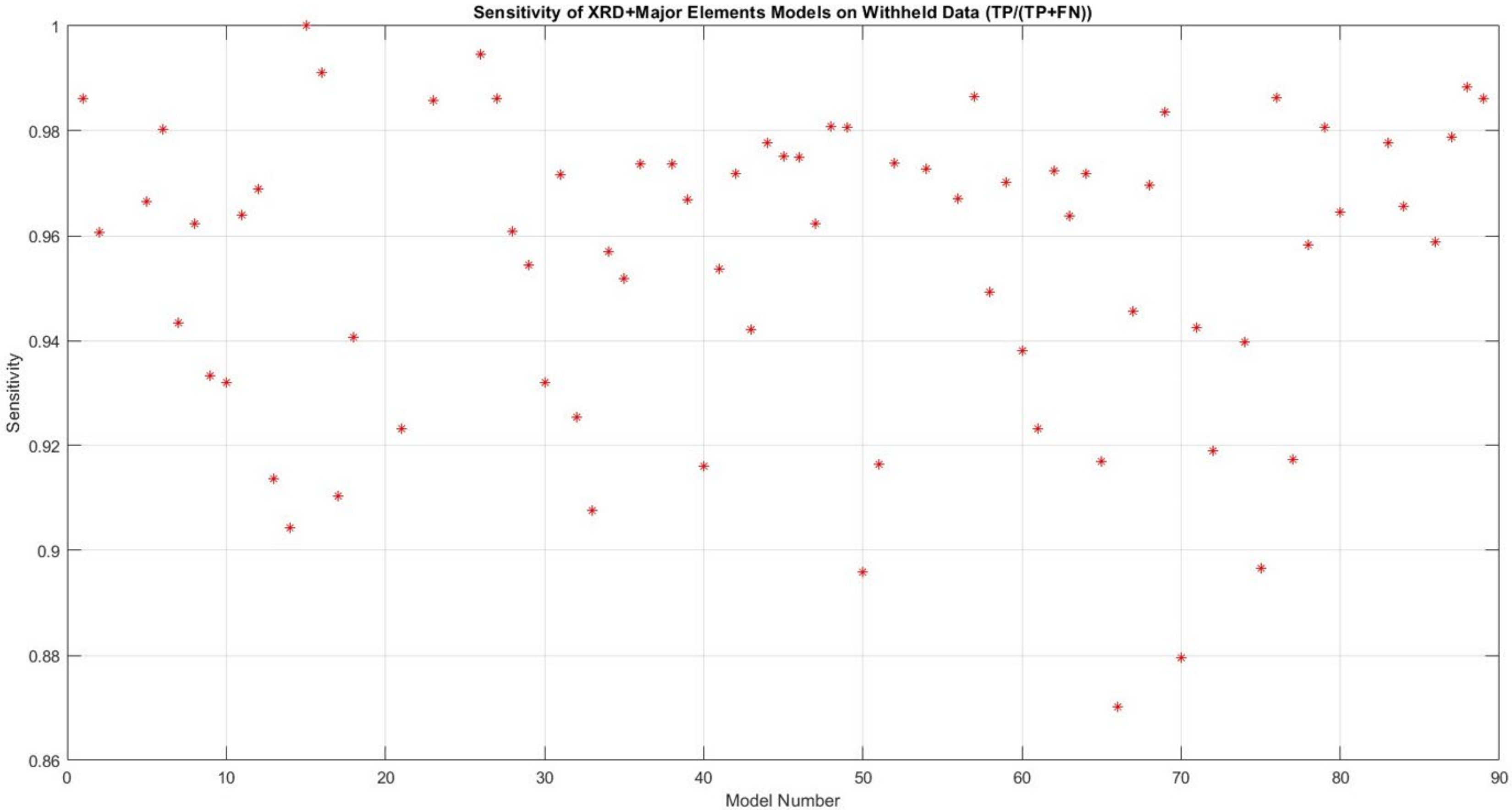

Figure 13. Sensitivity of XRD + Major Elements models on withheld generated data.
A majority of the models had a sensitivity of over 96% and were able to correctly identify the samples containing the target phase. Another chunk of models performed in the 90-96% range, while four models performed with a specificity below 90%. Those four phases were magnesite (50), pyrite (66), hypersthene (70), and sillimanite (75).

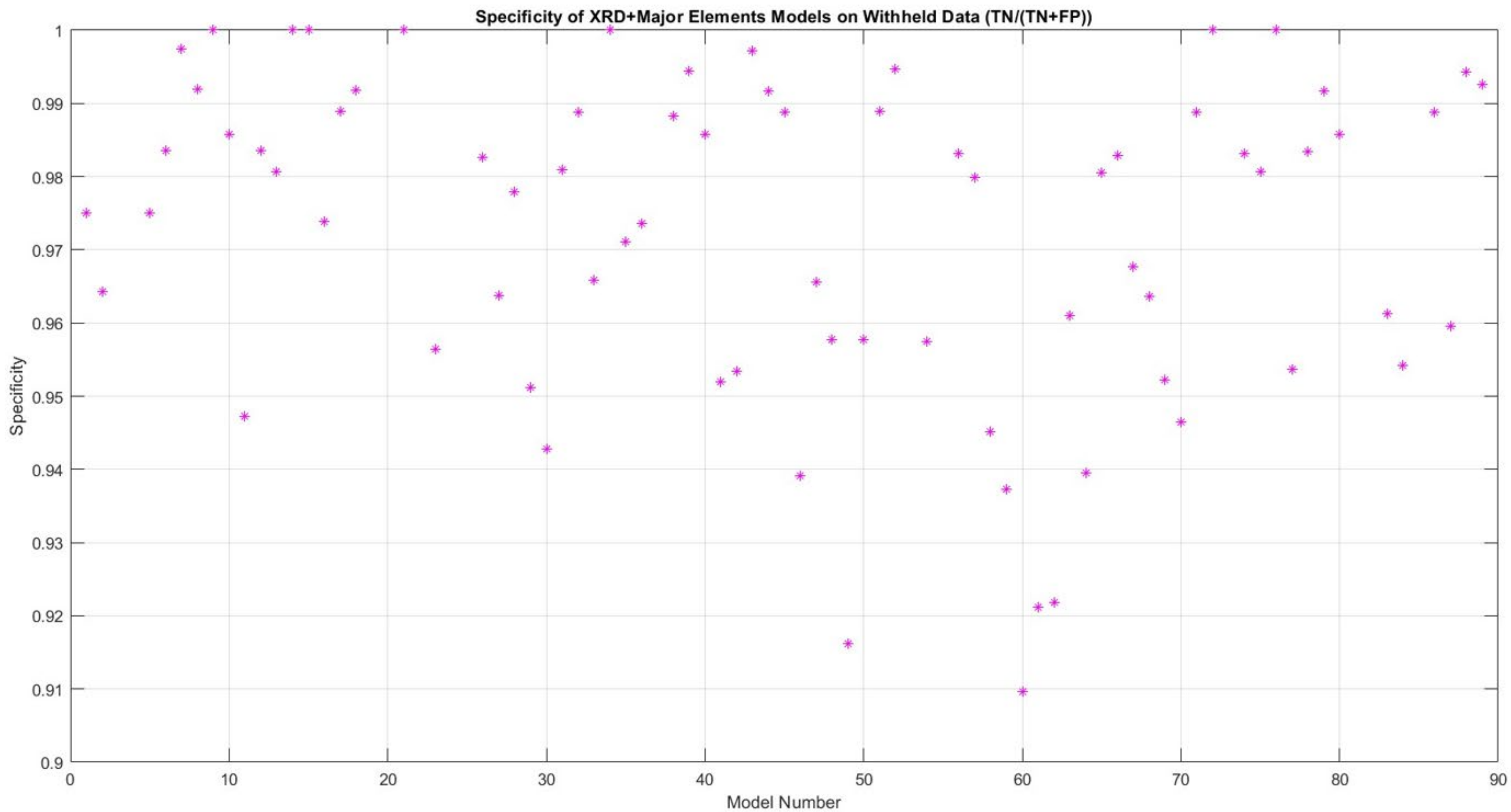

Figure 14. Specificity of XRD + Major Elements models on withheld generated data.
In general, all models were able to correctly identify the samples where the target phase was absent, with specificities close to or above 94%. The four models with lower specificities were sanidine (49), anorthite (60), bytownite (61), and labradorite (62).

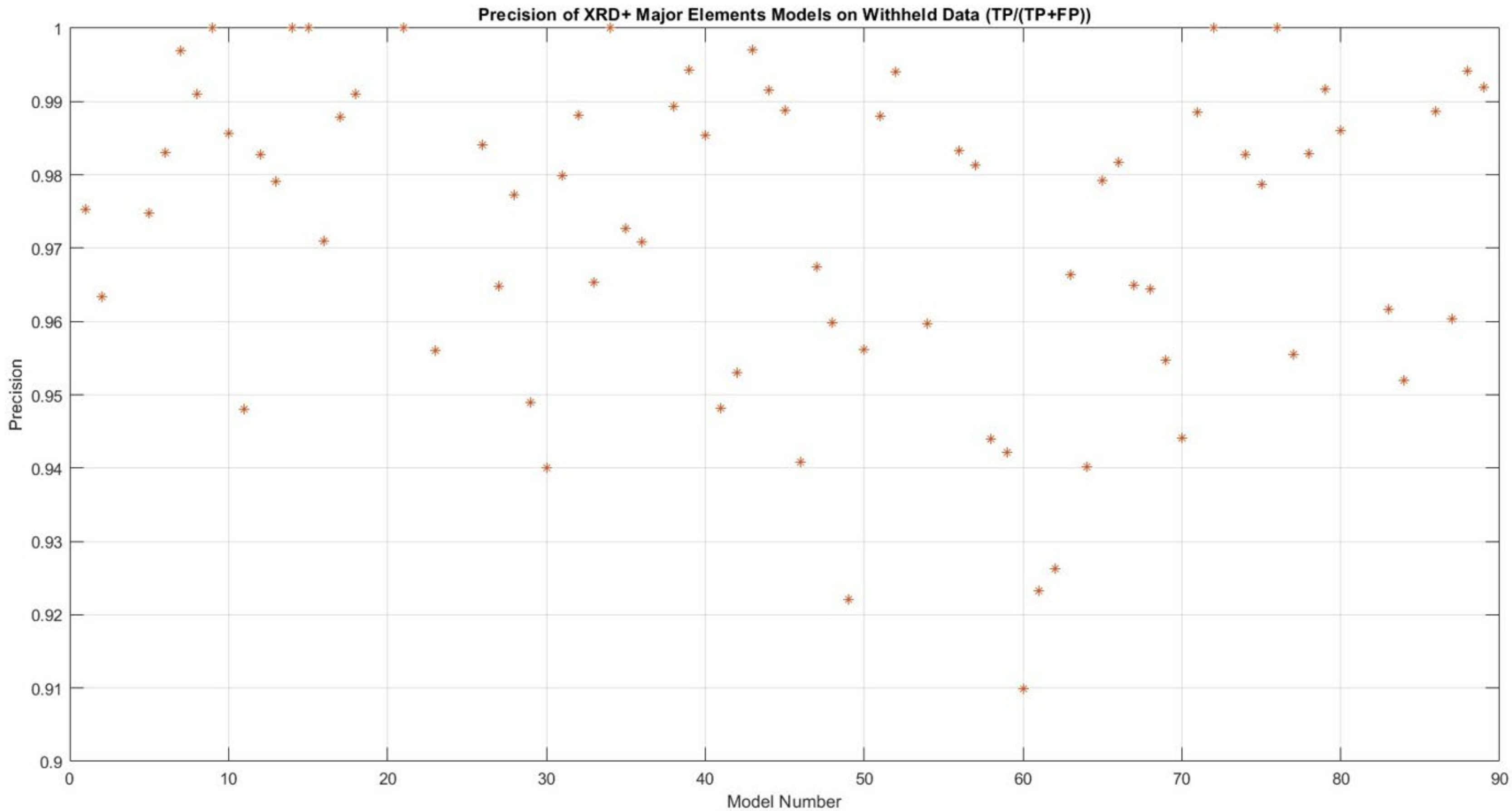

Figure 15. Precision of XRD + Major Elements models on withheld generated data.
All but 4 models had a precision higher than 94%, including many that were above 97%. The four models that performed worse were sanidine (49), anorthite (60), bytownite (61), and labradorite (62). Overall, the false positive rate was small for all models.

Table 24. XRD + Major Elements Models – Results on Withheld Data								
Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Alunite	1	0.9806	0.0194	0.9861	0.9750	0.9753	351	9
							5	355
Actinolite	2	0.9625	0.0375	0.9607	0.9643	0.9634	351	13
							14	342
Tremolite	5	0.9708	0.0292	0.9666	0.9751	0.9747	352	9
							12	347
Analcime	6	0.9819	0.0181	0.9803	0.9836	0.9831	359	6
							7	348
Anatase	7	0.9722	0.0278	0.9435	0.9974	0.9969	383	1
							19	317
Andalusite	8	0.9778	0.0222	0.9623	0.9920	0.9910	372	3
							13	332
Anglesite	9	0.9681	0.0319	0.9333	1.0000	1.0000	375	0
							23	322
Anhydrite	10	0.9583	0.0417	0.9321	0.9858	0.9856	347	5
							25	343
Ankerite	11	0.9556	0.0444	0.9639	0.9472	0.9481	341	19
							13	347
Aragonite	12	0.9764	0.0236	0.9689	0.9836	0.9828	360	6
							11	343
Arsenopyrite	13	0.9472	0.0528	0.9136	0.9806	0.9791	354	7
							31	328
Barite	14	0.9528	0.0472	0.9042	1.0000	1.0000	365	0
							34	321
Bassanite	15	1.0000	0.0000	1.0000	1.0000	1.0000	611	0
							0	109
Biotite_1m	16	0.9819	0.0181	0.9911	0.9738	0.9710	372	10
							3	335
Calcite	17	0.9500	0.0500	0.9104	0.9890	0.9878	359	4
							32	325
Celestine	18	0.9667	0.0333	0.9407	0.9918	0.9911	363	3
							21	333
Cinnabar	21	0.9625	0.0375	0.9233	1.0000	1.0000	368	0
							27	325
Cordierite	23	0.9708	0.0292	0.9858	0.9564	0.9560	351	16
							5	348
Diaspore	26	0.9889	0.0111	0.9947	0.9827	0.9841	340	6
							2	372

Dickite	27	0.9750	0.0250	0.9861	0.9638	0.9648	346 5	13 356
Dolomite	28	0.9694	0.0306	0.9608	0.9780	0.9772	355 14	8 343
Dolomite_Fe_rich	29	0.9528	0.0472	0.9544	0.9512	0.9490	351 16	18 335
Epidote	30	0.9375	0.0625	0.9320	0.9428	0.9400	346 24	21 329
Fluorapatite	31	0.9764	0.0236	0.9716	0.9810	0.9799	361 10	7 342
Fluorite	32	0.9569	0.0431	0.9254	0.9888	0.9882	354 27	4 335
Forsterite	33	0.9361	0.0639	0.9076	0.9659	0.9653	340 34	12 334
Galena	34	0.9792	0.0208	0.9569	1.0000	1.0000	372 15	0 333
Almandine	35	0.9611	0.0389	0.9519	0.9711	0.9727	336 18	10 356
Grossular	36	0.9736	0.0264	0.9737	0.9735	0.9708	368 9	10 333
Gypsum	38	0.9806	0.0194	0.9736	0.9883	0.9893	337 10	4 369
Halite	39	0.9806	0.0194	0.9669	0.9944	0.9943	355 12	2 351
Hematite	40	0.9500	0.0500	0.9160	0.9858	0.9854	346 31	5 338
Illite_1M_RM30	41	0.9528	0.0472	0.9536	0.9520	0.9481	357 16	18 329
Illite_2M1_SG4	42	0.9625	0.0375	0.9718	0.9534	0.9530	348 10	17 345
Ilmenite	43	0.9694	0.0306	0.9421	0.9972	0.9971	356 21	1 342
Jarosite_Mex	44	0.9847	0.0153	0.9777	0.9917	0.9915	359 8	3 350
Kaolinite_ Dry_Branch	45	0.9819	0.0181	0.9751	0.9889	0.9888	355 9	4 352
Anorthoclase	46	0.9569	0.0431	0.9749	0.9391	0.9409	339 9	22 350
Intermediate_ Microcline	47	0.9639	0.0361	0.9623	0.9656	0.9675	337 14	12 357
Ordered_ Microcline	48	0.9694	0.0306	0.9808	0.9577	0.9598	340 7	15 358

Sanidine	49	0.9486	0.0514	0.9807	0.9162	0.9221	328 7	30 355
Magnesite	50	0.9264	0.0736	0.8959	0.9577	0.9561	340 38	15 327
Magnetite	51	0.9528	0.0472	0.9164	0.9889	0.9880	357 30	4 329
Marcasite	52	0.9847	0.0153	0.9739	0.9947	0.9941	373 9	2 336
Muscovite_2M1	54	0.9653	0.0347	0.9728	0.9575	0.9597	338 10	15 357
Natrolite	56	0.9750	0.0250	0.9670	0.9831	0.9832	350 12	6 352
Phlogopite_2M1	57	0.9833	0.0167	0.9866	0.9798	0.9813	340 5	7 368
Albite	58	0.9472	0.0528	0.9493	0.9452	0.9440	345 18	20 337
Andesine	59	0.9542	0.0458	0.9702	0.9373	0.9421	329 11	22 358
Anorthite	60	0.9236	0.0764	0.9380	0.9096	0.9098	332 22	33 333
Bytownite	61	0.9222	0.0778	0.9233	0.9211	0.9233	327 28	28 337
Labradorite	62	0.9472	0.0528	0.9724	0.9218	0.9263	330 10	28 352
Oligoclase_NC	63	0.9625	0.0375	0.9638	0.9610	0.9663	320 14	13 373
Oligoclase_Norway	64	0.9556	0.0444	0.9719	0.9396	0.9402	342 10	22 346
Prehnite	65	0.9486	0.0514	0.9169	0.9805	0.9793	352 30	7 331
Pyrite	66	0.9250	0.0750	0.8703	0.9829	0.9817	344 48	6 322
Augite	67	0.9569	0.0431	0.9456	0.9677	0.9649	359 19	12 330
Diopside	68	0.9667	0.0333	0.9697	0.9636	0.9644	344 11	13 352
Hedenbergite	69	0.9681	0.0319	0.9835	0.9522	0.9547	339 6	17 358
Hypersthene	70	0.9125	0.0875	0.8795	0.9465	0.9441	336 44	19 321
Pyrrhotite	71	0.9653	0.0347	0.9425	0.9887	0.9885	351 21	4 344

Quartz	72	0.9597	0.0403	0.9190	1.0000	1.0000	362 29	0 329
Rutile	74	0.9611	0.0389	0.9397	0.9831	0.9828	349 22	6 343
Sillimanite	75	0.9389	0.0611	0.8966	0.9807	0.9787	355 37	7 321
Silver	76	0.9931	0.0069	0.9864	1.0000	1.0000	352 5	0 363
Sphalerite	77	0.9347	0.0653	0.9173	0.9536	0.9556	329 31	16 344
Spinel	78	0.9708	0.0292	0.9582	0.9834	0.9829	355 15	6 344
Strontianite	79	0.9861	0.0139	0.9807	0.9916	0.9916	355 7	3 355
Sulfur	80	0.9750	0.0250	0.9646	0.9858	0.9861	348 13	5 354
Titanite	83	0.9694	0.0306	0.9777	0.9612	0.9616	347 8	14 351
Tourmaline	84	0.9597	0.0403	0.9656	0.9542	0.9520	354 12	17 337
Zircon	86	0.9736	0.0264	0.9588	0.9888	0.9887	352 15	4 349
K-feldspars	87	0.9692	0.0308	0.9788	0.9596	0.9604	499 11	21 509
Plagioclases	88	0.9913	0.0087	0.9883	0.9943	0.9941	525 6	3 506
All Feldspars	89	0.9894	0.0106	0.9861	0.9926	0.9920	533 7	4 496

4.2.4. All Models – XRD + Composition (All Predictors)

The accuracy of predictions on generated data slightly increased compared to the other two types of models (Table 25, Figures 16-19). Overall accuracy for most composition models was above 94%, with only five models below 94% compared to eight models for major element models and six models for XRD-only models. Most models had an increase in precision to above 90%, but some individual plagioclase feldspar models suffered a decrease in their precision when composition variables were included.

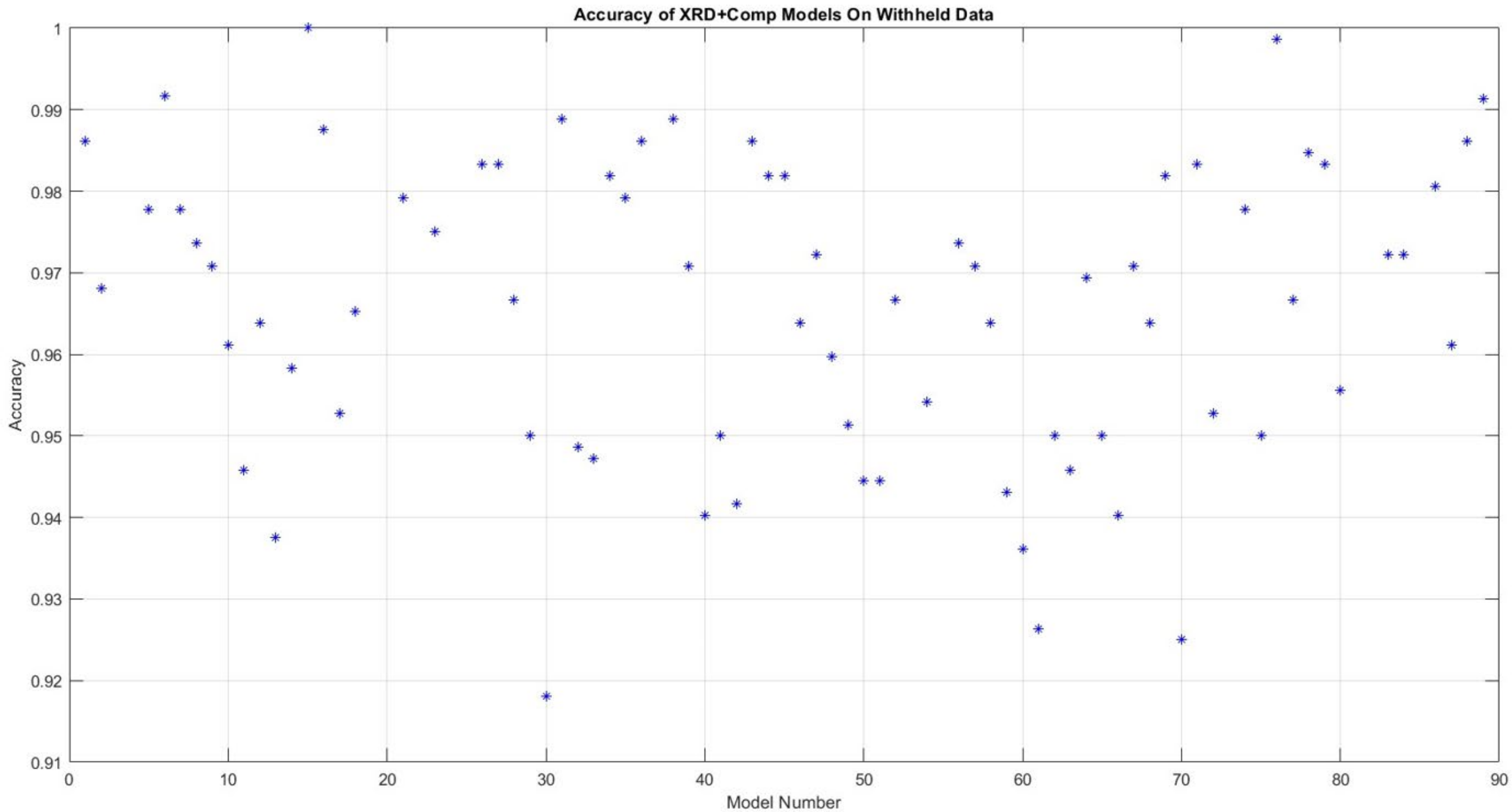

Figure 16. Accuracy of XRD + All Elements models on withheld generated data.
Overall, all models were very accurate on the withheld data with over 91% accuracy. Only 5 models had lower than 94% accuracy – arsenopyrite (13), epidote (30), anorthite plagioclase (60), bytownite plagioclase (61), and hypersthene pyroxene (70). Most models clustered between 94-99% accuracy.

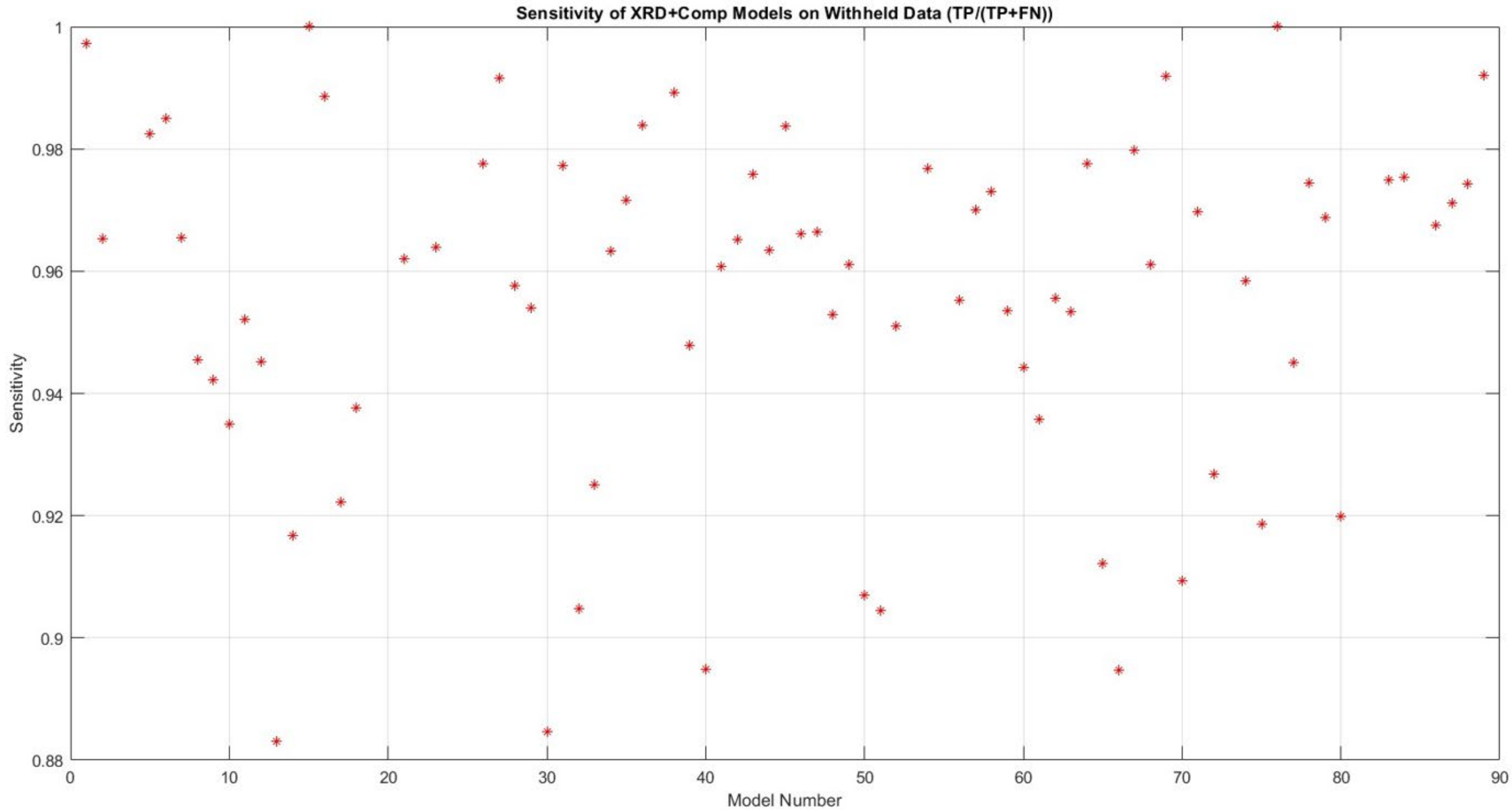

Figure 17. Sensitivity of XRD + All Elements models on withheld generated data.
The sensitivity of the models was the lowest overall accuracy measure for the models with all composition data. While most models still performed with higher than 96% sensitivity, there were some that dropped below 90% on the withheld data. Some phases in particular that were misclassified more than others were arsenopyrite (13), epidote (30), fluorite (32), hematite (40), magnesite (50), magnetite (51), prehnite (65), pyrite (66), and hypersthene pyroxene (70).

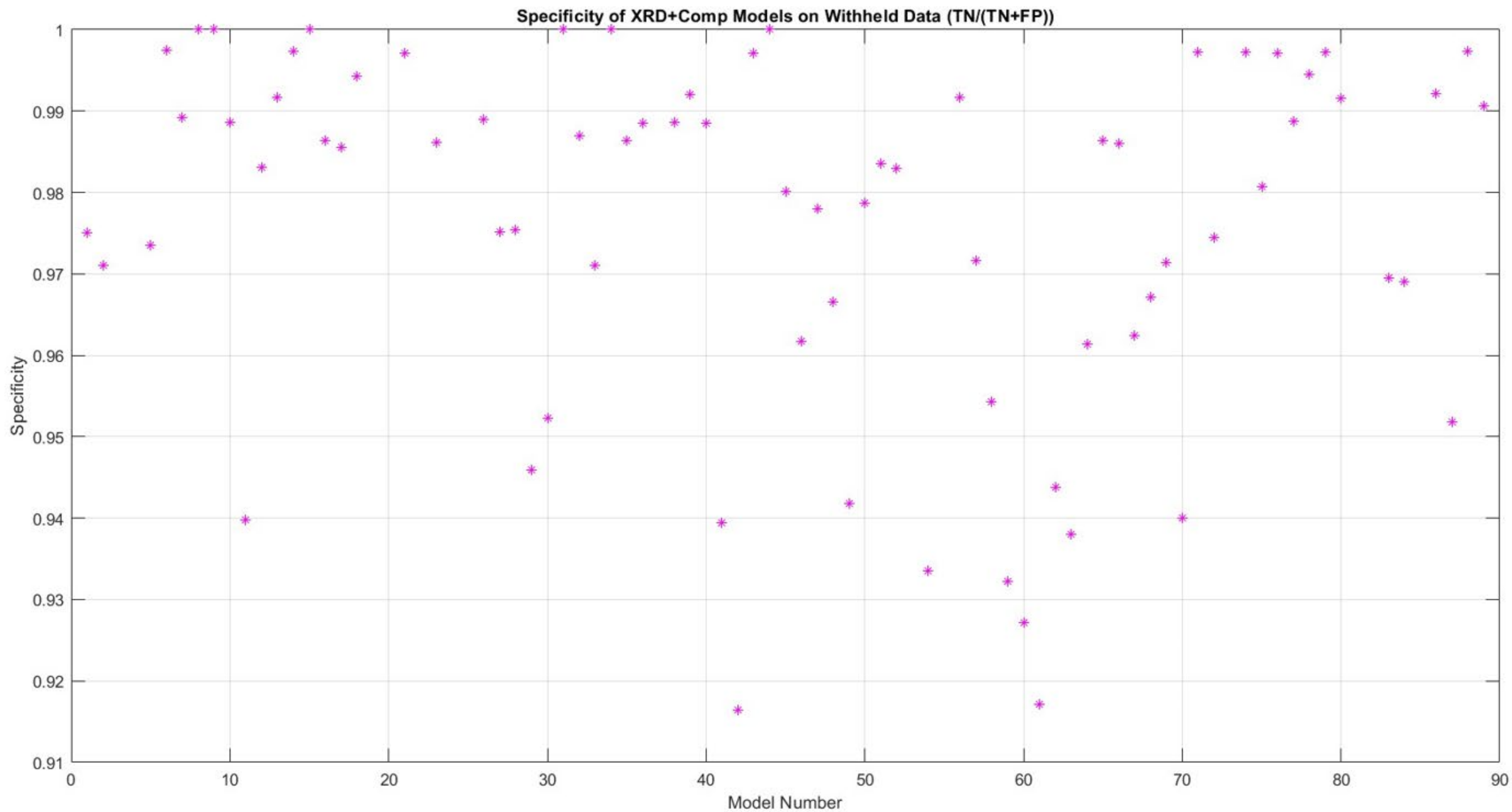

Figure 18. Specificity of XRD + All Elements models on withheld generated data.
All models were able to predict the absence of the target phase well, with specificity over 91%. Most models were above 98% specificity. The lowest specificity models were illites (41, 42), sanidine (49), muscovite (54), plagioclase feldspars (58-64), and hypersthene pyroxene (70).

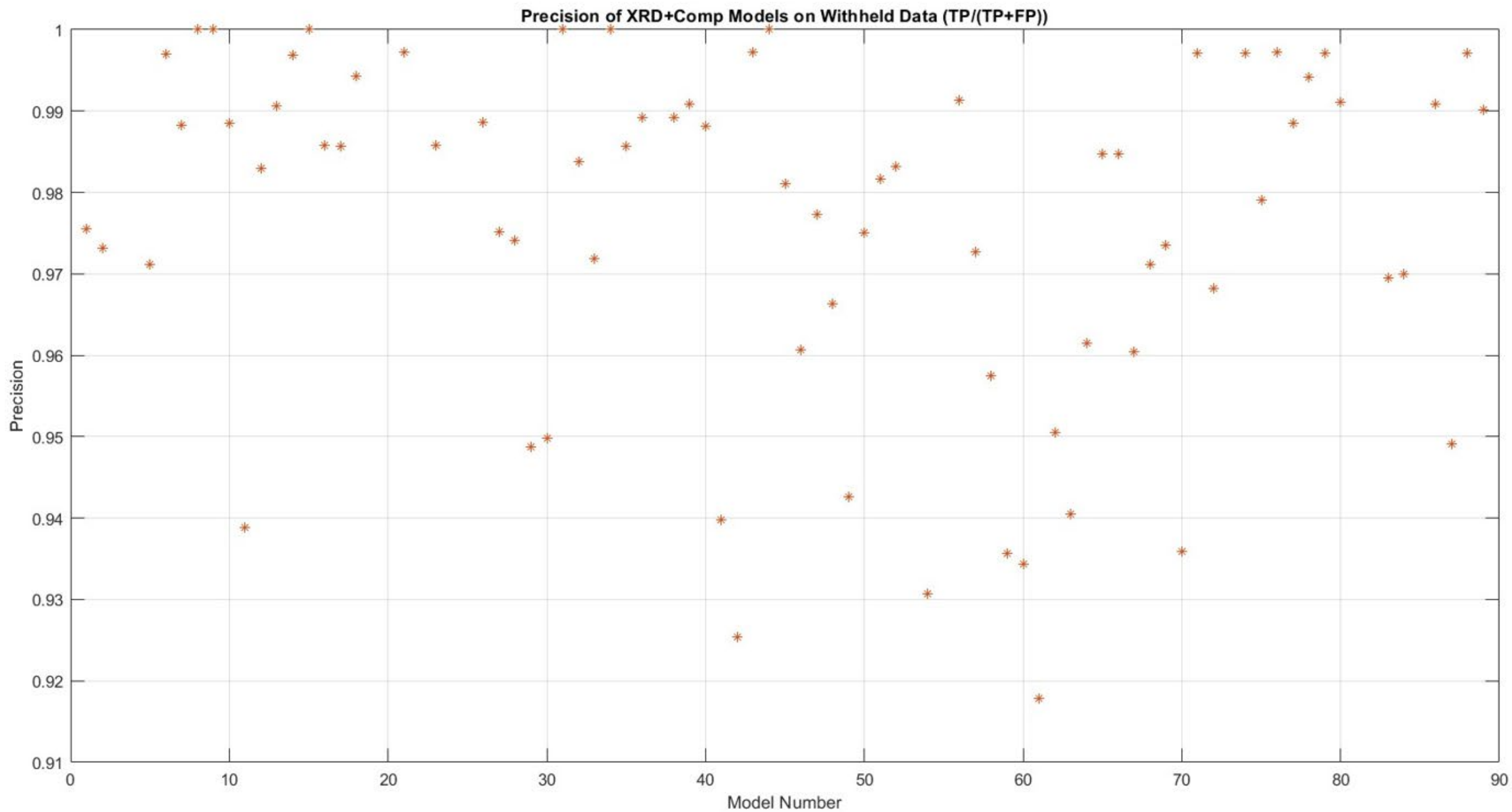

Figure 19. Precision of XRD + All Elements models on withheld generated data.
All models performed with higher than 91% precision, indicating a low number of false positives. Most models were above 95% on their precision. Models that had slightly lower precision rates were ankerite (11), illites (41, 42), muscovite (54), plagioclase feldspars (59-63), and hypersthene (70). The alkali feldspar group model (88) had lower precision than the other feldspar group models.

Table 25. XRD + All Elements Models – Results on Withheld Data

Name	Model ID	Accuracy Measures					Confusion Matrix	
		Accuracy	Misclass.	Sensitivity	Specificity	Precision		
Alunite	1	0.9861	0.0139	0.9972	0.9750	0.9755	351	9
							1	359
Actinolite	2	0.9681	0.0319	0.9653	0.9710	0.9731	335	10
							13	362
Tremolite	5	0.9778	0.0222	0.9825	0.9735	0.9712	367	10
							6	337
Analcime	6	0.9917	0.0083	0.9849	0.9974	0.9970	387	1
							5	327
Anatase	7	0.9778	0.0222	0.9655	0.9892	0.9882	371	0
							19	330
Andalusite	8	0.9736	0.0264	0.9456	1.0000	1.0000	371	0
							19	330
Anglesite	9	0.9708	0.0292	0.9421	1.0000	1.0000	357	0
							21	342
Anhydrite	10	0.9611	0.0389	0.9350	0.9886	0.9885	347	4
							24	345
Ankerite	11	0.9458	0.0542	0.9521	0.9397	0.9389	343	22
							17	338
Aragonite	12	0.9639	0.0361	0.9452	0.9831	0.9829	349	6
							20	345
Arsenopyrite	13	0.9375	0.0625	0.8830	0.9917	0.9906	358	3
							42	317
Barite	14	0.9583	0.0417	0.9167	0.9973	0.9969	371	1
							29	319
Bassanite	15	1.0000	0.0000	1.0000	1.0000	1.0000	594	0
							0	126
Biotite_1m	16	0.9875	0.0125	0.9886	0.9864	0.9858	363	5
							4	348
Calcite	17	0.9528	0.0472	0.9223	0.9856	0.9857	342	5
							29	344
Celestine	18	0.9653	0.0347	0.9377	0.9943	0.9943	349	2
							23	346
Cinnabar	21	0.9792	0.0208	0.9620	0.9972	0.9972	351	1
							14	354
Cordierite	23	0.9750	0.0250	0.9639	0.9861	0.9858	355	5
							13	347
Diaspore	26	0.9833	0.0167	0.9776	0.9890	0.9887	359	4
							8	349

Dickite	27	0.9833	0.0167	0.9916	0.9752	0.9752	354 3	9 354
Dolomite	28	0.9667	0.0333	0.9576	0.9754	0.9741	357 15	9 339
Dolomite_Fe_rich	29	0.9500	0.0500	0.9539	0.9459	0.9488	332 17	19 352
Epidote	30	0.9181	0.0819	0.8846	0.9522	0.9499	339 42	17 322
Fluorapatite	31	0.9889	0.0111	0.9773	1.0000	1.0000	367 8	0 345
Fluorite	32	0.9486	0.0514	0.9048	0.9870	0.9838	379 32	5 304
Forsterite	33	0.9472	0.0528	0.9251	0.9711	0.9719	336 28	10 346
Galena	34	0.9819	0.0181	0.9633	1.0000	1.0000	366 13	0 341
Almandine	35	0.9792	0.0208	0.9717	0.9864	0.9856	362 10	5 343
Grossular	36	0.9861	0.0139	0.9839	0.9885	0.9892	343 6	4 367
Gypsum	38	0.9889	0.0111	0.9892	0.9886	0.9892	346 4	4 366
Halite	39	0.9708	0.0292	0.9478	0.9920	0.9909	372 18	3 327
Hematite	40	0.9403	0.0597	0.8949	0.9885	0.9881	345 39	4 332
Illite_1M_RM30	41	0.9500	0.0500	0.9608	0.9394	0.9397	341 14	22 343
Illite_2M1_SG4	42	0.9417	0.0583	0.9651	0.9164	0.9254	318 13	29 360
Ilmenite	43	0.9861	0.0139	0.9758	0.9971	0.9973	347 9	1 363
Jarosite_Mex	44	0.9819	0.0181	0.9635	1.0000	1.0000	364 13	0 343
Kaolinite_ Dry_Branch	45	0.9819	0.0181	0.9837	0.9801	0.9810	345 6	7 362
Anorthoclase	46	0.9639	0.0361	0.9661	0.9617	0.9607	352 12	14 342
Intermediate_ Microcline	47	0.9722	0.0278	0.9664	0.9780	0.9773	355 12	8 345
Ordered_ Microcline	48	0.9597	0.0403	0.9529	0.9666	0.9663	347 17	12 344

Sanidine	49	0.9514	0.0486	0.9610	0.9418	0.9426	340 21 14 345
Magnesite	50	0.9444	0.0556	0.9070	0.9787	0.9750	368 8 32 312
Magnetite	51	0.9444	0.0556	0.9045	0.9835	0.9817	358 6 34 322
Marcasite	52	0.9667	0.0333	0.9511	0.9830	0.9831	346 6 18 350
Muscovite_2M1	54	0.9542	0.0458	0.9767	0.9335	0.9307	351 25 8 336
Natrolite	56	0.9736	0.0264	0.9552	0.9917	0.9913	360 3 16 341
Phlogopite_2M1	57	0.9708	0.0292	0.9700	0.9717	0.9727	343 10 11 356
Albite	58	0.9639	0.0361	0.9730	0.9543	0.9574	334 16 10 360
Andesine	59	0.9431	0.0569	0.9536	0.9322	0.9357	330 24 17 349
Anorthite	60	0.9361	0.0639	0.9443	0.9271	0.9344	318 25 21 356
Bytownite	61	0.9264	0.0736	0.9358	0.9171	0.9178	332 30 23 335
Labradorite	62	0.9500	0.0500	0.9555	0.9438	0.9505	319 19 17 365
Oligoclase_NC	63	0.9458	0.0542	0.9534	0.9380	0.9405	333 22 17 348
Oligoclase_Norway	64	0.9694	0.0306	0.9777	0.9613	0.9615	348 14 8 350
Prehnite	65	0.9500	0.0500	0.9122	0.9864	0.9847	362 5 31 322
Pyrite	66	0.9403	0.0597	0.8947	0.9861	0.9848	354 5 38 323
Augite	67	0.9708	0.0292	0.9798	0.9625	0.9605	359 14 7 340
Diopside	68	0.9639	0.0361	0.9610	0.9672	0.9711	324 11 15 370
Hedenbergite	69	0.9819	0.0181	0.9919	0.9713	0.9735	339 10 3 368
Hypersthene	70	0.9250	0.0750	0.9093	0.9401	0.9359	345 22 32 321
Pyrrhotite	71	0.9833	0.0167	0.9697	0.9972	0.9972	356 1 11 352

Quartz	72	0.9528	0.0472	0.9268	0.9745	0.9682	382 10 24 304
Rutile	74	0.9778	0.0222	0.9583	0.9972	0.9971	359 1 15 345
Sillimanite	75	0.9500	0.0500	0.9185	0.9808	0.9790	357 7 29 327
Silver	76	0.9986	0.0014	1.0000	0.9972	0.9973	352 1 0 367
Sphalerite	77	0.9667	0.0333	0.9451	0.9888	0.9885	352 4 20 344
Spinel	78	0.9847	0.0153	0.9744	0.9946	0.9942	366 2 9 343
Strontianite	79	0.9833	0.0167	0.9688	0.9973	0.9971	367 1 11 341
Sulfur	80	0.9556	0.0444	0.9199	0.9916	0.9911	355 3 29 333
Titanite	83	0.9722	0.0278	0.9749	0.9695	0.9695	350 11 9 350
Tourmaline	84	0.9722	0.0278	0.9753	0.9690	0.9700	344 11 9 356
Zircon	86	0.9806	0.0194	0.9675	0.9921	0.9909	379 3 11 327
K-feldspars	87	0.9611	0.0389	0.9711	0.9519	0.9492	356 18 10 336
Plagioclases	88	0.9861	0.0139	0.9742	0.9973	0.9971	370 1 9 340
All Feldspars	89	0.9913	0.0087	0.9921	0.9907	0.9901	530 5 4 501

4.2.5. Feldspar Groups

As the feldspar minerals were commonly the lowest performing phases, each feldspar group was combined into one model to attempt to improve prediction accuracy of the group as a whole. Grouping the phases into one model for alkali feldspars and one for plagioclases predicted with consistently high accuracy on the withheld data for all model types but showed only a slight improvement in accuracy on the real mixture samples for the XRD-only models. As feldspar XRD patterns typically have many low intensity peaks as opposed to a few large peaks, a stepwise analysis may improve the accuracy of identifying feldspars. If phases that are certain to be in the model are removed, the difference pattern between the original pattern and the removed phases can be inputted back into the models to be able to identify phases with lower intensity peaks.

5. DISCUSSION

The feature ranking process and the performance of the machine learning models in classifying the phases from both generated and real mixtures give some interesting insights into what features of an XRD pattern are the most important for machine learning models to correctly identify the phase. These insights can be used to help train novice analysts to know what to look for in XRD analysis to better identify mineral phases.

5.1. XRD Pattern

The feature ranking process gave key insights into what the algorithms deem important for distinguishing a phase from the rest of an XRD pattern. In all phases examined, the locations of the major peaks of its pattern are key to identifying its presence in the sample, whether you’re a human analyst or a machine. This was especially the case for more simple patterns like calcite and quartz. How the pattern changes around the peak, represented by the first and second derivative of the pattern, is also helpful for the machine learning models to identify the phase in a pattern. Using the first and second derivative of the pattern is an advantage that the machine learning models would have on human analysts as those patterns may not be as easy for a human analyst to interpret.

5.2. Elemental Composition

Composition predictors can help with predictions for some minerals, as evidenced in the feature ranking for halite. With the generated validation set, including just the major elements increased the halite model's accuracy from 97.36% with just the XRD pattern to 98.06%. Tourmaline model performance also improved in every accuracy measure with the addition of the trace elements – accuracy increased from 95.97% for pattern only models to 97.22% with all predictors, sensitivity increased from 95.63% to 97.53%, specificity from 96.33% to 96.90%, and precision from 96.42% to 97.00%. This suggests that similar minerals that contain trace elements or rarer elements in their compositions would also improve from addition chemical information when identifying them in XRD analysis with the machine learning models. Novice analysts can also learn from this to incorporate results such as XRF data to improve their results.

5.3. Scores – How Certain Was the Model?

When predicting new data with a random forest model in MATLAB, there is an additional output called scores that gives the fraction of decision trees that voted for each class (absent=0, present=1). This measure can be used to represent the probability of the presence of the target phase in the sample, according to the model. The different calcite models I tested had a wide range of scores for the first 14 samples of the real mixture test data set as shown in Table 21. The only samples that consistently had about 50/50 scores were the three Reynold's Cup 10 samples (the first three samples listed). This makes sense however, as the source of these XRD patterns informed me they did not have the best quality XRD measurement due to complications in the sample preparation process. The scores for the quartz models tested during the sample number selection experiment also provided insight, as samples that were misclassified usually had scores that were also close to 50/50. This is encouraging, as a score close to a 50/50 score for misclassified or poorly prepared samples could help prompt the analyst to investigate further to determine the presence of the phase, even if the model in the end classified the phase as absent.

5.3.1. Weight Percentage Effect

When investigating the various XRD-only quartz models, the weight percentage of quartz in the sample had a large effect on the accuracy of the models and the certainty of their scores as seen in Figure 20. Where the weight percentage of quartz was between 2.5-15%, the models were less certain of whether the mineral was present or not. For 1% and under, the models usually predicted that quartz was absent. This may be because at small weight percentages, the intensities of the peaks are smaller and easily overshadowed by other minerals. Without additional composition predictors or applying an iterative analysis that subtracts the high percentage minerals, it seems the models will misidentify the presence of the phase at lower weight percentages. However, the addition of the scores in analysis results can inform an analyst as to the quality of the results and if further investigation is necessary.

Weight percentage might have additional effects on the data that would be worth further investigation. Future research could test if weight percentage has effects on other aspects of the models such as feature importance, for example if the important features for prediction change when there is a lower weight percentage of the target mineral. Datasets with different ranges of percentages of the target mineral can be created to test how the feature ranking as well as the model's predictions change with weight percentage.

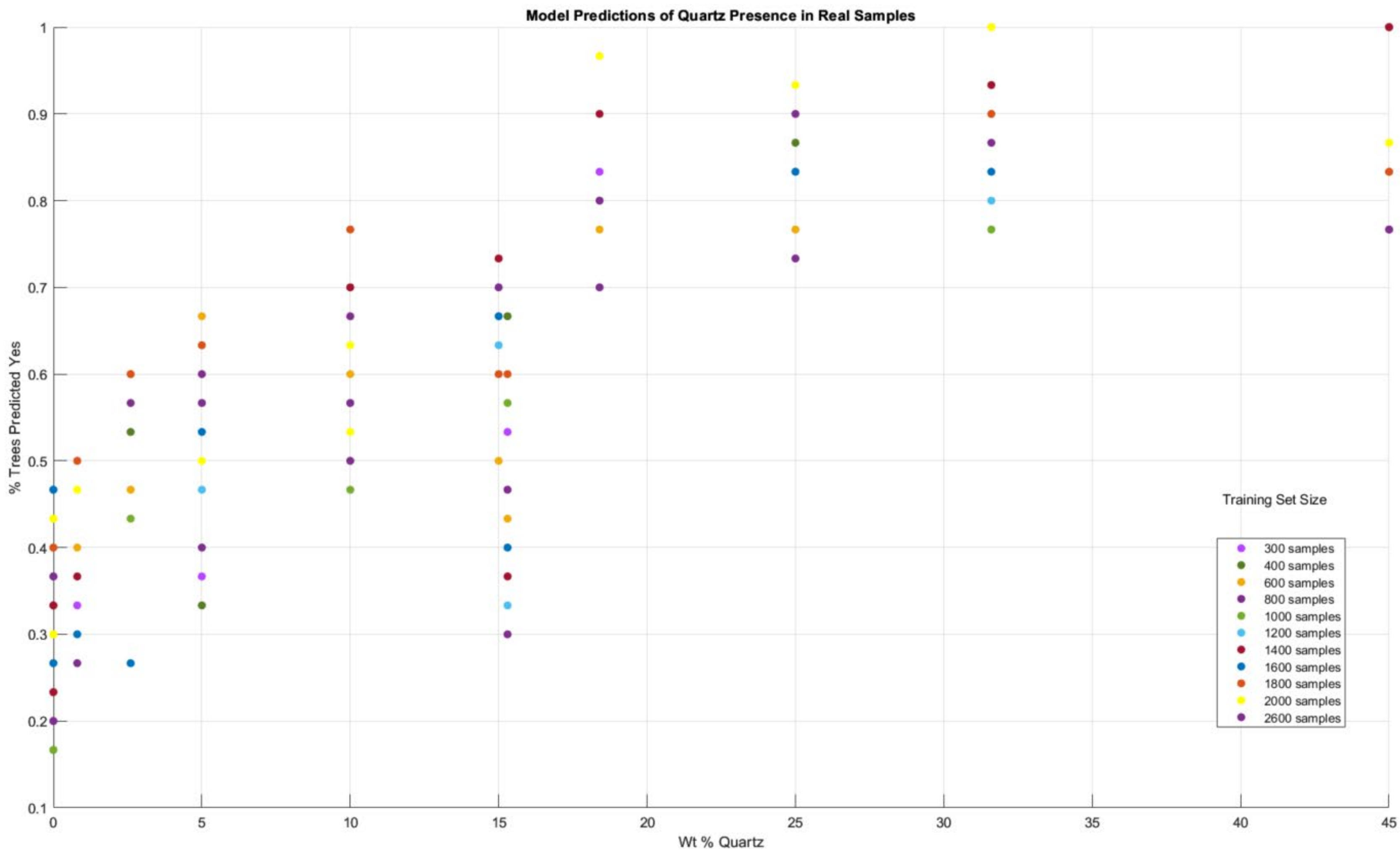

Figure 20. Models' predictions of quartz with varying training set sizes
For each model trained on a different size of training set, this shows the fraction of the trees in the ensemble that predicted the mineral was present. For models above 17% quartz, most of the trees in each model correctly predicted the presence of quartz. More uncertainty comes between 1-15% quartz content.

5.4. Clay Minerals

Since XRD patterns of clay minerals can vary dramatically due to particle swelling and rotational disorder, most were not included in sample generation. Phases that were included were kaolinite, dickite, illite 1M, and illite 2M. However, most of the real mixture test data did include other clays like smectite that were not included in the generated samples. Figure 21 shows the variance measured by residuals from the plot of the scores for the various quartz models plotted against the weight percentage of clays and amorphous material in the sample. The variance is tightest around 0% clays and is greatest around 20% clays. This suggests that the inclusion of clays can affect the models I trained by making the results more inconsistent in their predictions.

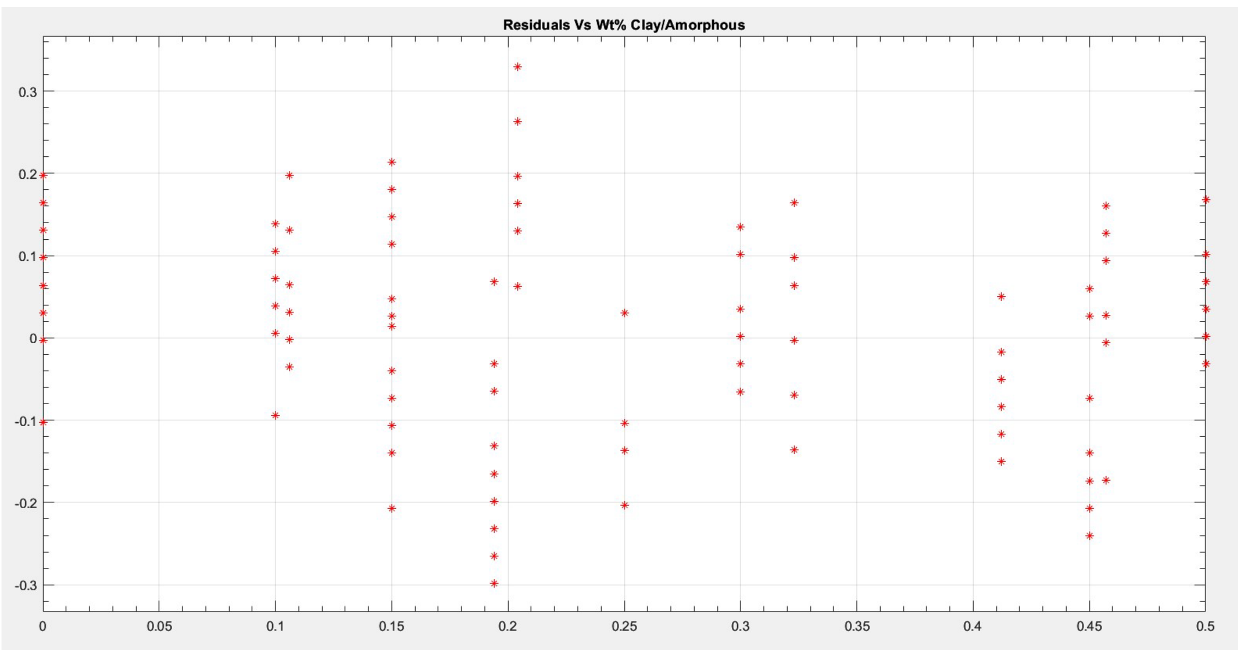

Figure 21. Quartz regression residuals vs weight fractions of clays in test samples
A linear regression model was fit to the plot in Figure 20, and residuals were measured. These residuals are plotted versus the weight fractions of clays and amorphous material included in the test samples. The variance in the model predictions is greatest around 20% clays and smallest at 0%.

These results show the absence of clays in the training data is one of the biggest limitations of the models. In future research, this can be remedied by generating samples that have clays included. Other machine learning models can also be developed to learn the effect that clays and swelling have on an XRD pattern. A model, such as a neural network, can be trained on pairs of synthetic mixtures – one without the clay and one with the clay included. The model can then learn the difference between the two patterns and be able to potentially: 1. identify the clay phases present, and 2. subtract the clay effect from the pattern so other mineral phases can be more easily identified. These types of models would be better equipped to handle the identification and effects of clay minerals in XRD patterns.

6. CONCLUSION

XRD analysis is a common method to quantify the relative amounts of phases in a sample. However, current methods can be difficult for analysts, especially novice analysts, to use due to the large amount of user input required by XRD analytical software to identify phases to include and make an accurate analysis. To alleviate this problem, I developed machine learning models to mimic the choices and geologic knowledge of expert analysts. In order to prevent overfitting, train an accurate machine learning model, and obtain enough samples of rare phases, I needed a very large data set of XRD patterns, and the weight percentages of the phases included. I created a synthetic data set of XRD patterns with analogues of expert geologic knowledge through MATLAB code. These samples included 2-15 phases and their elemental composition data. Random forests of decision trees proved to be the most consistent and accurate model to correctly identify the phases in the samples. Most phases were able to be identified in real mixtures with greater than 90% accuracy.

Although clays and lower weight percentages can influence the certainty of the models' predictions, the inclusion of elemental composition data has the potential to mitigate this effect. Future research would need to train machine learning models on the effects of clays in an XRD pattern to increase the accuracy and precision of these models. Models that test how weight percentage of the target mineral affects feature ranking and the model’s predictive ability would give more insight into the limitations of machine learning and how to improve its accuracy.

REFERENCES

Alpaydin, E. (2014). Introduction to Machine Learning (Third ed., p. 338). Cambridge, MA: The MIT Press.

Bish, D.L. and Howard, S.A. (1988) Quantitative phase analysis using the Rietveld method. Journal of Applied Crystallography, 21, 86-91.

Ciaburro, Giuseppe. (2017). MATLAB for Machine Learning. Packt Publishing.

Downs, R.T. and Hall-Wallace, M. (2003) The American Mineralogist Crystal Structure Database. American Mineralogist 88, 247-250.

Eberl, D.D. (2003) User's Guide to RockJock—A Program for Determining Quantitative Mineralogy from Powder X-Ray Diffraction Data. In: U.S.G. Survey Ed., Boulder, Colorado.

Fenner, M.E. (2020). Machine Learning with Python for Everyone. Addison-Wesley.

Jenkins, R., and Snyder, R.L. (1996). Introduction to X-ray Powder Diffractometry. John Wiley & Sons, Inc., pp. 89-94.

Knox, S.W. (2018) Machine Learning: A Concise Introduction, 320 p. Wiley, Hoboken, New Jersey.

Kumar, R. (2019). Machine Learning Quick Reference. Birmingham, UK: Packt Publishing.

MATLAB. (2023). "Ensemble Algorithms." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/stats/ensemble-algorithms.html#btfwpd3

MATLAB. (2023). "Feature Selection and Feature Transformation Using Classification Learner App." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/stats/feature-selection-and-feature-transformation.html

MATLAB. (2023). "One-way analysis of variance (anova1)." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/stats/anova1.html

MATLAB. (2023). "Kruskal-Wallis test (kruskalwallis)." MATLAB Documentation, MathWorks, Inc., 2023, www.mathworks.com/help/stats/kruskalwallis.html#btv4oqy-9

MATLAB. (2023). "MRMR feature ranking (fscmrmr)." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/stats/fscmrmr.html#mw_6e50c940-81bb-4df0-8ce0-7aa97f4ed0aa

MATLAB. (2023). "Select Data for Classification." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/stats/select-data-and-validation-for-classification-problem.html

MATLAB. (2023). "Tall Arrays for Out-of-Memory Data." MATLAB Documentation, MathWorks, Inc., www.mathworks.com/help/matlab/import_export/tall-arrays.html

Pecharsky, V.K., Zavalij, P.Y. (2009). Fundamentals of Powder Diffraction and Structural Characterization of Materials. (Second ed.), Springer, pp. 151-200.

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow (Second ed.). Birmingham, UK: Packt Publishing.

Srodon, J., et al. (2001). "Quantitative X-ray Diffraction Analysis of Clay-Bearing Rocks from Random Preparations." Clays and Clay Minerals, vol. 49, no. 6, pp. 514-28.

Zheng, A. and Casari, A. (2018) Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, 218 p. O'Reilly, Boston.

Zhou, X., Liu, D., Bu, H., Deng, L., Liu, H., Yuan, P., Du, P., and Song, H. (2018) "XRD-based quantitative analysis of clay minerals using reference intensity ratios, mineral intensity factors, Rietveld, and full pattern summation methods: A critical review". Solid Earth Sciences, 3, 16-29.