

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

Detector Pixel Calibration of Time-of-flight Neutron Diffractometers Accelerated by Machine Learning



Albert Song
Ke An

06/2025

DOCUMENT AVAILABILITY

Online Access: US Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via <https://www.osti.gov>.

The public may also search the National Technical Information Service's [National Technical Reports Library \(NTRL\)](#) for reports not available in digital format.

DOE and DOE contractors should contact DOE's Office of Scientific and Technical Information (OSTI) for reports not currently available in digital format:

US Department of Energy
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@osti.gov
Website: www.osti.gov

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Neutron Scattering Division

**DETECTOR PIXEL CALIBRATION OF TIME-OF-FLIGHT NEUTRON
DIFFRACTOMETERS ACCELERATED BY MACHINE LEARNING**

Author(s)

**Albert Song
Ke An**

June 2025

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

ABSTRACT	1
1. INTRODUCTION	2
2. TIME FOCUSING PROCESS	2
3. PIXEL DIFFRACTION CONSTANT CALIBRATION BY MACHINE LEARNING.....	5
4. SUMMARY	8
ACKNOWLEDGEMENT	9
REFERENCES.....	9

ABSTRACT

Modern time-of-flight neutron diffractometers at spallation neutron source are equipped with two dimensional detectors with fine pixelations. The flight path of neutrons from the moderator to the sample and to the detector needs to be precisely calibrated at detector pixel level using standard powders so the diffraction data from all the detector pixels can be correctly time-focused to produce high resolution diffraction peaks. The number of pixels can reach to millions which makes a single-pixel calibration process time-consuming, or even impossible, with conventional fitting routine. Here we presented a machine learning aided calibration process via a “training and predict” process by training machine learning models with the relations between the individual pixel time-of-flight diffraction pattern and fitted diffraction constant. The training models take a portion of the available pixels to predict the diffraction constants precisely and rapidly for massive pixel diffraction patterns.

1. INTRODUCTION

At the Spallation Neutron Source (SNS), neutrons are produced via spallation process when high energy proton pulses generated from an accelerator hitting liquid mercury filled in a steel target chamber. The neutron pulses travel through moderators, instrument guide systems and beam slits, then scatter with samples. And the scattered neutrons are collected by neutron detectors by recording the detector pixel number and arriving time to the specific detector pixel, i.e. neutron events, including the unique pulse number for each neutron pulse, literally, the birth time stamp when the proton pulse hits the target. Although the neutron pulses are generated with wide energies or wavelengths, the specific neutron wavelength can be converted from velocity via Planck-Einstein relation. The velocity of neutron from spallation process can be calculated from the time-of-flight (TOF) and the traveling distance from the moderator to the detector. To accurately determine TOF, given a high time resolution data acquisition system for example, the 10^{-7} s or 100 nano second at the SNS, the distance between the sample to the individual single pixel needs to be precisely determined and calibrated with diffraction line from standard calibrant, e.g. standard Si powders. The engineering design and survey of the detector locations provide the first approximation of the pixel distance from the ideal sample position, a measurement of long-time exposure of the standard calibrant is conducted for refining the pixel level distances before the overall combined pattern is generated and further single pattern Rietveld refinement is conducted to determine high order instrument parameters. Nowadays, two-dimensional neutron detectors used at a modern spallation neutron facility have finer pixelation and wider coverage. An instrument can be equipped with detectors consist of millions of pixels, the smallest detection unit in a detector module. To calibrate the distance of the detector pixel from the sample position, the pixel level Rietveld refinement of its diffraction pattern is not practical because it is too time consuming; single peak fitting or cross correlation method to calculate the relative difference of the pixel distance speed up the process without refining the full pattern, and work well for the instrument with thousands of pixels. However, the compute time adds up quickly when dealing with instruments with millions of pixels. A more robust and rapid method is needed to meet the advanced instrumentation. Machine learning (ML) provides a promising opportunity for processing large data set where a training model can be established by training a subset data with pixel diffraction pattern and the distance from sample or diffraction constant and then predict rapidly the overall pixel diffraction constants. ML is a fast approach because it can run on GPU when needed while the prediction process at individual pixel level is similar and can run concurrently. In this paper, we used a diffraction calibration data from a TOF diffractometer instrument at the SNS to explore the possibility of training ML models and conduct pixel level distance calibration without the need of any mathematical fitting or Rietveld refinement. A few ML training models were compared, and the performance of the ML approach is presented. The training models take a portion of the available pixels and successfully predict precisely and rapidly the diffraction constants for massive pixel diffraction patterns.

2. TIME FOCUSING PROCESS

As mentioned above, a neutron event is recorded when a detectable neutron arrives at a detector pixel. The pair of TOF of the neutron and the detector pixel identification number are registered as one neutron event, which are in real 32-bit type. For a given pixel, diffraction the histogram looks like one in Figure 1, in which a diffraction pattern of diamond powders is represented. Due to the change of the locations of the detector pixels, there will be slight drift of the diffraction pattern, as a result, a contour plot of diffraction pattern from many detector pixels will show peak shifts as shown in Figure 2, in which a representation of up to 2048 diffraction patterns from pixels is shown. The shifts depend on the location of the pixels and the Bragg angles, and it could change with large range for the whole instrument with wide detector coverage in both vertical and horizontal directions. The shift needs to be aligned well via a so-called time focusing process to produce a useful whole diffraction pattern of the powder. It is worth

being noted here, a diffractometer may have to produce multiple diffraction pattern by grouping nearest banks because the instrument broadening is heavily depending on Bragg angle. Having multiple patterns could avoid deteriorating diffraction line width from high angle detectors.

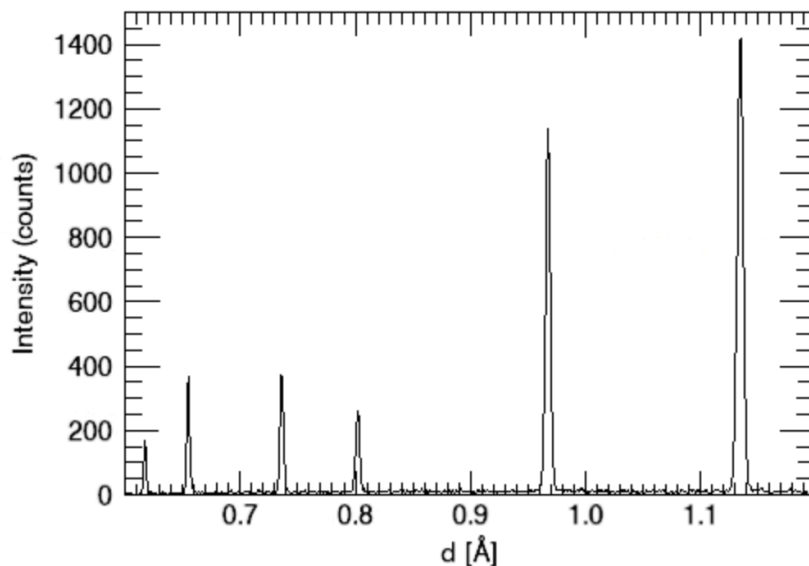


Figure 1 Neutron diffraction of diamond powder of a single detector pixel. Data was histogrammed directly on the collected TOFs and convert the value to d spacing for the purpose of showing.

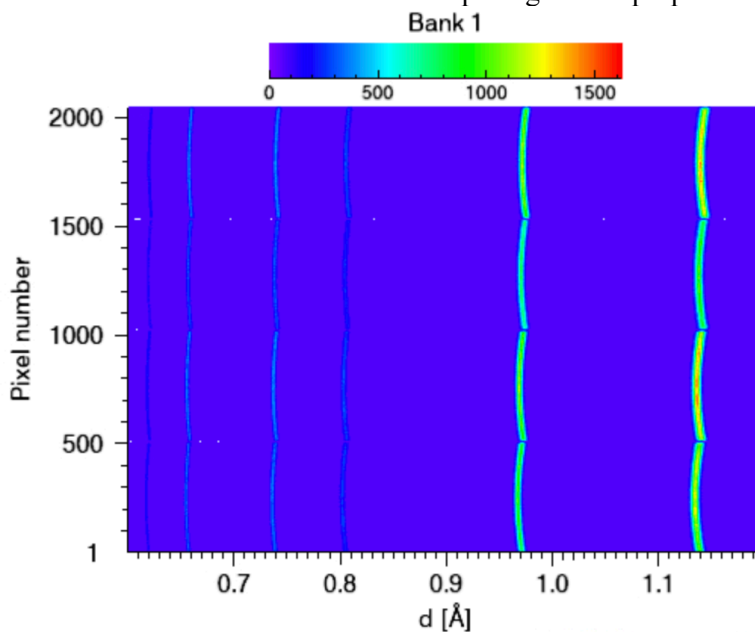


Figure 2 The contour plot of detector pixel diffraction data from four ^3He detector tubes. The data were histogrammed directly on TOFs and converted in d-spacing by a constant diffraction constant, i.e. before time-focusing.

The time focusing process requires adjusting pixel neutron events TOF with respect to a reference pixel, so all the pixels data can be added up and produce the histogram of sample data in one detector bank. The reference pixel is the time focusing center and can be defined from either a physical or virtual location. The time adjustment or time offset can be determined by knowing the pixel locations which are firstly approximated by the engineering design and alignment survey and later being calibrated by measuring a diffraction calibrant. For one individual detector pixel in TOF neutron diffraction, the relationship between the d-spacing for a particular powder line and its TOF is [1]:

$$Tof_i = DIFC_i \cdot d + DIFA_i \cdot d^2 + ZERO \quad (1)$$

in time focusing with in a detector bank the significant factor to the time adjustment is the diffraction constant $DIFC_i$, which is a factor from detector pixel location and geometry:

$$DIFC_i = 252.816 \cdot 2 \sin \Theta \left(L_1 + \sqrt{L_{2,i}^2 + \frac{L_{3,i}^2}{16}} \right) \quad (2)$$

where Θ is the Bragg angle, L_1 is the primary flight path, L_2 is sample to detector pixel center distance and L_3 is the height of the detector pixel. Among the pixels, they share same L_1 , and possess similar L_3 values, thus the changes of $DIFC$ for each pixel is a function of its physical location which define the distance of L_2 and Bragg angle. Event mode time focusing algorithm is carried out directly on event data by the first order TOF correction. When neglecting small $DIFA_i \cdot d^2$ and $ZERO$ values, the equation (1) above can be simplified as:

$$Tof_i = DIFC_i \cdot d \quad (3)$$

applying arithmetic logarithm on the equation, we have:

$$\log(Tof_i) = \log(DIFC_i) + \log(d) \quad (4)$$

the logarithmic time offset of pixel to a reference pixel is given by:

$$Tof_{offset,i} = \log(Tof_i) - \log(Tof_0) = \log(DIFC_i) - \log(DIFC_0) \quad (5)$$

if a time offset table is known for a set of detector pixels, the TOF time focusing can be easily calculated as below:

$$\log(Tof_{c,i}) = \log(Tof_i) - Tof_{offset,i} \quad (6)$$

Therefore, to obtain the time offset table for time focusing, we need to determine $DIFC_i$ for each pixel with known lattice d in equation (3). A variant of such process is to covert TOF of single pixels to d space, and histogram the neutron events in d then covert back to TOF for most data analysis software. This is doable but adding conversion steps consume compute time but most of the time it is negligible for small data set.

Calibrating the $DIFC_i$ can be done by fitting single or multiple diffraction lines in the diffraction pattern as shown in Figure 1. The ratios of the peak positions in TOF and known diffraction d-spacing values are the $DIFC_i$ according to equation (3). One requirement in doing this fitting is considering the peak profile, which is usually convolution of Pseudo-Voigt and back-to-back exponential. Another approach used widely at the SNS is to find the logarithmic time offset of pixels from conducting a cross

correlation between the pixel diffraction patterns. The cross correlated results are quickly fitted by a gaussian profile, the center peak of the shifted lags is converted to the logarithmic time offset for time focusing. The cross-correlation method could utilize the full pattern without the need to fit the full pattern; this presumably rapidly produces time focusing table with enough accuracy as compared to the single peak or full pattern fitting approach. The method has been actively used on VULCAN instrument [2] by measuring a strong diamond powder scatter and being reduced by VDRIVE software [3].

3. PIXEL DIFFRACTION CONSTANT CALIBRATION BY MACHINE LEARNING

Nowadays, instruments are equipped with detectors with higher pixelation numbers because of finer detector pixel size and the wider detector coverage. For example, in just one bank of the VULCAN instrument, it has over 81,920 individual pixels that each detect their own diffraction pattern. The commonly used two VULCAN banks have total 163,840 pixels. Many more pixels are expected as more banks are being added to the instrument. To be noted here, in a ^3He detector tube, the pixel size along the tube can be reduced simply in data acquisition software and the density of pixelation increases. Single crystal instruments with finer pixelation and more complete detector coverage can exceed easily more than 1 million pixels. It is obvious that the pixel level calibration of the diffraction constant requires faster and efficient method in addition to powerful compute power. The process of calibration at pixel level is similar but with large quantities therefore machine learning could play a key role in such type of challenges.

In ML, a model is trained with prepared data of observation and response variable, so the model could find patterns and make predictions of new response variable with new observations. In this case, the select subset of detector pixel diffractions are the observations, and the corresponding diffraction constants are the response variables. For the initial training with small subset data, the matching diffraction constant can be predetermined by conventional methods such as single peak fitting or cross correlation on the small set of data at first. Once a model is trained to be reliable enough, it can then make response variable prediction of diffraction constants with given pixel diffraction patterns. When a full set of calibration is needed due to change of instrument condition such as a new experiment configuration with change of slits or collimations, the established training model does not need to be retrained but could provide fast and efficient instrument pixel calibration.

Here, specifically about a demonstration by using the example data of from bank 1 and 2 from VULCAN, we used Pandas, a Python data analysis library, to combine all pixel diffraction data files into one large CSV file/ matrix (pixel-number X time-bin-number) format, where each row was a pixel, while the columns were the time bins (~3000 with constant logarithmic bin width of 0.001), with the values within each cell corresponding to the intensity of a pixel at a time bin. We will use subset of these intensities as our predictors for our eventual ML to conduct the training and prediction task as mentioned above.

On the training models, we investigated multiple different machine learning models to achieve optimal performance. Specifically, we investigated the performances of AdaptiveBoost [4], GradientBoosting [5], and XGBoost [6]. Those are all ensemble machine learning methods that build strong predictive models by combining the outputs of several weaker learners. They differ in performances with respect to compute speed, precision and on a few other aspects. And they are readily implementable in python (via importing `sklearn.ensemble.AdaBoostClassifier`, `sklearn.ensemble.GradientBoostingClassifier`, and `xgboost`, respectively).

To compare these, we first establish a baseline by creating a naïve model and evaluating the mean absolute error (MAE). The MAE is based on the difference of the ML predicted diffraction constants and the calibrated diffraction constants. The target of acceptable MAE is set to be a few unit values of the diffraction constant as a general practice in diffraction instrument calibration. We then tuned the hyperparameters for each of the different ML models, with the loss function being the MAE. To do this, we first performed a coarse grid search, with a wide range of hyperparameters to approximate the ideal hyperparameters to minimize our loss function. Then, once that was done, we performed a finer grid search to finalize the hyperparameters of each model. As for the real data, we set a total intensity threshold to exclude ‘bad’ pixels with low intensity which are normally close to the end of the detector tubes. Those ‘bad’ pixels are also masked in use of generating sample diffraction data.

For ML model training, we employed several hardware setups. Our primary hardware setup was the Remote Analysis Cluster (analysis.sns.gov) at the SNS, a remote desktop service that employs very powerful computers with multiple CPUs as well as over 500 GB of RAM. Additionally, we used the ORNL DGX platform, a GPU cluster with multiple CPUs as well as over 1 TB of RAM to train the XGBoost as we could utilize its GPU cluster to speed up our model fitting. Finally, we compared using these remote hardware setups with using a Dell Latitude 7520 laptop with one CPU and 16 GB of RAM.

To establish a reliable training model, large training sizes are needed, and the prediction outcome will be more reliable with increasing training size. For each of the models we investigated, we tuned and trained them on a randomly sample 20% subset of the data. This was chosen as that would equate to around 30,000 samples (pixel data), and since we had around 3,000 (bins) predictors. We investigated this after finalizing our hyperparameters by fitting our AdaBoost model to differing sizes of training data, shown in Figure 3. As we see with using just 20% of the data as the training set, we can achieve an MAE of below 2 ms/Å, which meets the desired goodness prediction. Given the diffraction constant to be near 16300 ms/Å, the MAE is ~0.01% of the diffraction constant value. 0.01% is more than needed precision for DIFCs as the d spacing measurement accuracy is near 0.0001 Å. Having a smaller training set also helps us avoid potentially overfitting the data and 10% of training size is also acceptable. The improvement slow down with increase of the training size. Indeed, having a training set of 90% of the data, while yielding a lower MAE for the remaining 10% test set, could potentially result in a model that could be used for future prediction of full calibrations from new set of calibration measurement.

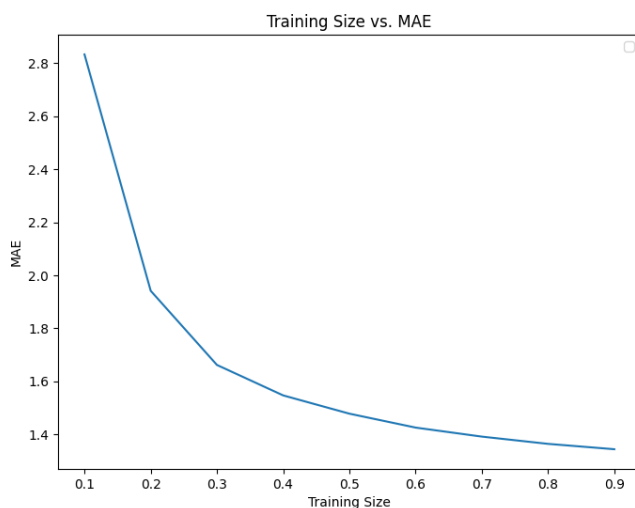


Figure 3 MAE dependency of training size as a proportion of entire set.

After tuning the ideal hyperparameters for each model, we fit each model on the training dataset and found the residuals for the test set. Figure 4 shows the residuals ($\text{ms}/\text{\AA}$) for each model plotted by pixel number.

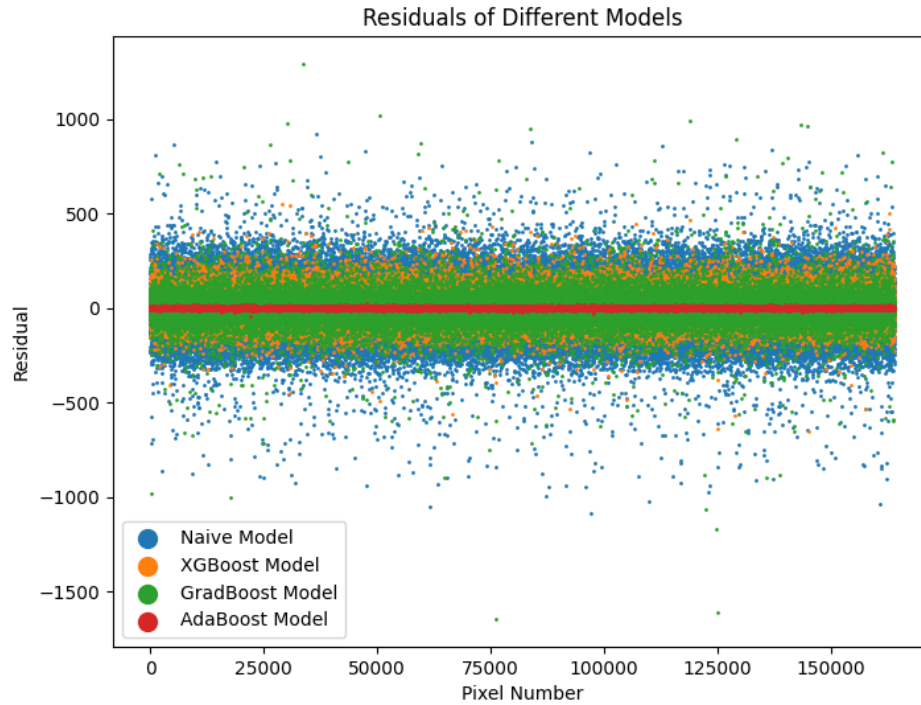


Figure 4 Residuals of models investigated vs. pixel number

We can clearly see that while XGBoost and GradBoost offer mild improvements over the naïve model, AdaBoost results in vastly superior predictive performance when compared to the other models. Additionally, the root mean square error (RMSE), which is a measure of variance of our predictions, was also quite low, meaning that the residuals are consistent over the entire validation set. Our detailed MAEs and RMSEs are listed below in Table 1.

Table 1: MAEs and RMSEs of models investigated.

Models	MAE($\text{ms}/\text{\AA}$)	RMSE ($\text{ms}/\text{\AA}$)
Naïve	95.98	125.99
XGBoost	39.30	55.41
GradBoost	25.60	49.12
AdaBoost	1.94	2.68

Using this method saves us time that would otherwise be spent either fitting or cross-correlating hundreds of thousands of peaks. This method is also flexible due to its generalized nature. If the training set is formatted correctly, we can train a model to recognize where the peak is to a high accuracy. This means that this method can be used at not only VULCAN, but it can be applied to other instruments with

similar time-of-flight calibration procedures. We have shown that using our model, we can predict the peaks of 120,000 sets of data in under five minutes with a very high accuracy.

However, this method is limited by the time it takes to tune and train a model. For tuning, a typical grid search involves fitting up to 300 candidate models. With each fit taking approximately 30 minutes even on server level hardware, time and resources quickly becomes a limiting factor. Even with 10 processes running simultaneously, tuning still takes upwards of 12 hours to perform. One of the reasons we chose a low training size was hardware limitations that we encountered while tuning. We initially used an 80% training size with respect to the entire dataset, but we encountered issues with memory, as we found that we were using upwards of 1 TB of RAM, which was infeasible with our hardware setups. Additionally, since we are using each distance bin as predictors, we must develop a new model every time we change our distance bins. Since each of these fits takes a significant amount of time and resources, we cannot implement real-time fitting. Finally, the models we investigated were largely within the scikit-learn Python library, which does not come with GPU-compatible algorithms. This does not let us leverage ORNL's powerful GPU clusters to speed up the process of training the models. Future work should be done to implement the models into other libraries that are GPU-compatible such as TensorFlow.

4. SUMMARY

In this work we successfully demonstrated that using machine learning approach, detector pixel calibration of diffraction constants can be speedily conducted. The example training models can use fraction of the data samples to obtain reliable training result and to precisely predict diffraction constants by just reading pixel-level diffraction spectra. It is a very convincing for large detector data processing, and because of its simplicity, operation in GPU is foreseen doable as well and time saving could be even more significant given large and fine pixelations of modern neutron detectors.

ACKNOWLEDGEMENT

This research used resources the Spallation Neutron Source, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory. A.S. thanks the support from Science Undergraduate Laboratory Internship (SULI) program sponsored by DOE Office of Science when he came from Northwestern University in summer 2023.

REFERENCES

1. Allen C. Larson, R.B.V.D., *General Structure Analysis System (GSAS)*. 2004, Los Alamos National Laboratory.
2. An, K., et al., *First In Situ Lattice Strains Measurements Under Load at VULCAN*. Metallurgical and Materials Transactions a-Physical Metallurgy and Materials Science, 2011. **42a**(1): p. 95-99.
3. An, K., *VDRIVE- Data Reduction and Interactive Visualization Software for Event Mode Neutron Diffraction*. 2012, Oak Ridge National Laboratory.
4. Freund, Y. and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 1997. **55**(1): p. 119-139.
5. Friedman, J.H., *Greedy function approximation: A gradient boosting machine*. Annals of Statistics, 2001. **29**(5): p. 1189-1232.
6. Chen, T.Q. and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*. Kdd'16: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016: p. 785-794.