# HVCM Anomaly Detection

Yasir Alanazi

Collaborators:
M. Schram, K. Rajput (Jefferson Lab)
D. Lu, M. Radaideh, C. Pappas, P. Ramuhalli (ORNL)

ORNL is managed by UT-Battelle, LLC for the US Department of Energy
JLab is managed by Jefferson Sc. Assoc., LLC for the US Department of Energy

# Outline

- Machine Learning-based Anomaly Detection
- HVCM Anomaly Detection at SNS
  - Introduction
  - Autoencoder & Variational Autoencoder
  - Data preparation
  - Single Module-based model
  - Multi Module-based model
- Model Evaluation
  - Loss Landscape visualization

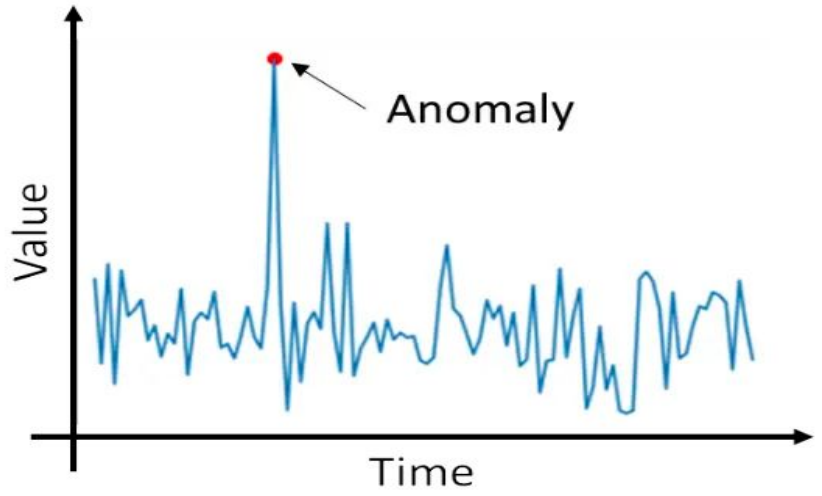
# Machine Learning-based Anomaly Detection

- Anomaly detection: is identifying data points that do not fit normal patterns. (aka outlier detection).

- Applications of anomaly detection include:
  - Fault detection in manufacturing.
  - Fraud detection in financial transactions.
  - Health Monitoring.
  - Transportation, ...etc.

- Machine Learning have been recently used in several anomaly detection applications.

- Why we need Machine Learning?
  - Huge amounts of data, i.e: texts, images, videos.
  - Data is often unstructured.




* Srivignesh R

# Machine Learning-based Anomaly Detection

- Anomaly detection using ML methods:
  - Supervised learning:
    - Use a complete set of normal & abnormal labels (i.e. Classification).
  - Unsupervised learning:
    - Detect anomalies in unlabeled datasets (assumptions: majority of the instances in the data set will be normal).
  - Semi-supervised learning:
    - Use a normal, labeled training data set to construct a model representing normal behavior.
    - Then use the model to detect anomalies by testing how likely the model is to generate any one instance encountered.
    - I.e. Autoencoders (AEs) & Variational Autoencoders (VAEs).

# Outline

- Machine Learning-based Anomaly Detection
- HVCM Anomaly Detection at SNS
  - Introduction
  - Autoencoder & Variational Autoencoder
  - Data preparation
  - Single Module-based model
  - Multi Module-based model
- Model Evaluation
  - Loss Landscape visualization

# HVCM Anomaly Detection at SNS

- The Spallation Neutron Source (SNS) facility delivers the most intense neutron beam in the world for scientific materials research.
- The SNS consists of 15 High Voltage Converter Modulators (HVCMs) which are used to convert 13.8 kVAC to up to 135 kV pulses at 60 Hz for 1.3 ms.
- The HVCMs occasionally experience failures which can result in a day or more of lost operation time.
- The HVCM is the second leading source of downtime for the SNS after the proton target system (Fig. 1).

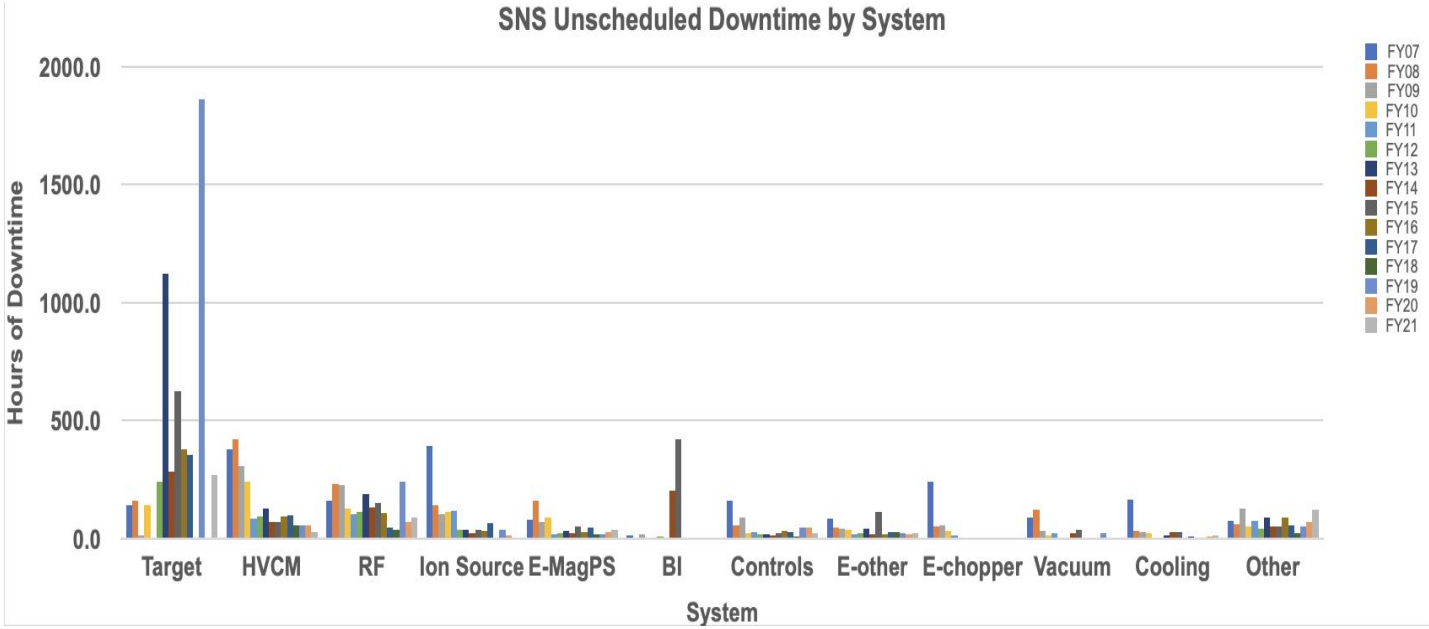



*Figure 1. Spallation neutron source downtime by system from fiscal year (FY) 2007 to 2019.*

*Courtesy to Chris Pappas, ORNL

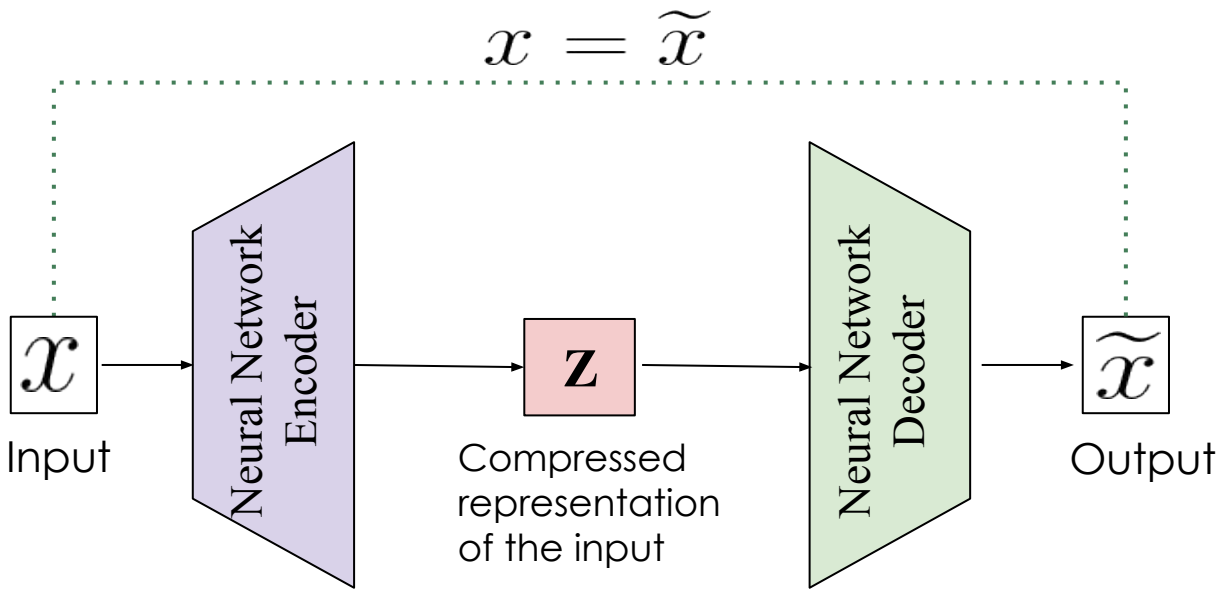
# HVCM Anomaly Detection at SNS

- **Goal**: Predict an upcoming machine failure before it occurs to improve the reliability of the HVCMs and reduce the down time for SNS facility.
- **How:** We use pulses leading to failure because we believe there is a sign about upcoming anomaly event before it happens.
- **Pipeline:**
  - Data Source: Experimental data
  - Data Preparation: Extract normal & Abnormal waveforms
  - ML Approach: Variational Autoencoder (VAE)
  - Training tools: Keras with Tensorflow backend
  - Results: Predict anomalies for several fault types
  - Model Evaluation: Loss landscape visualization

# What is Autoencoder (AE)

- Autoencoder is a class of machine learning that can be used to reconstruct data.
  - It was originally implemented as a nonlinear dimensionality reduction method (e.g, reduce images).

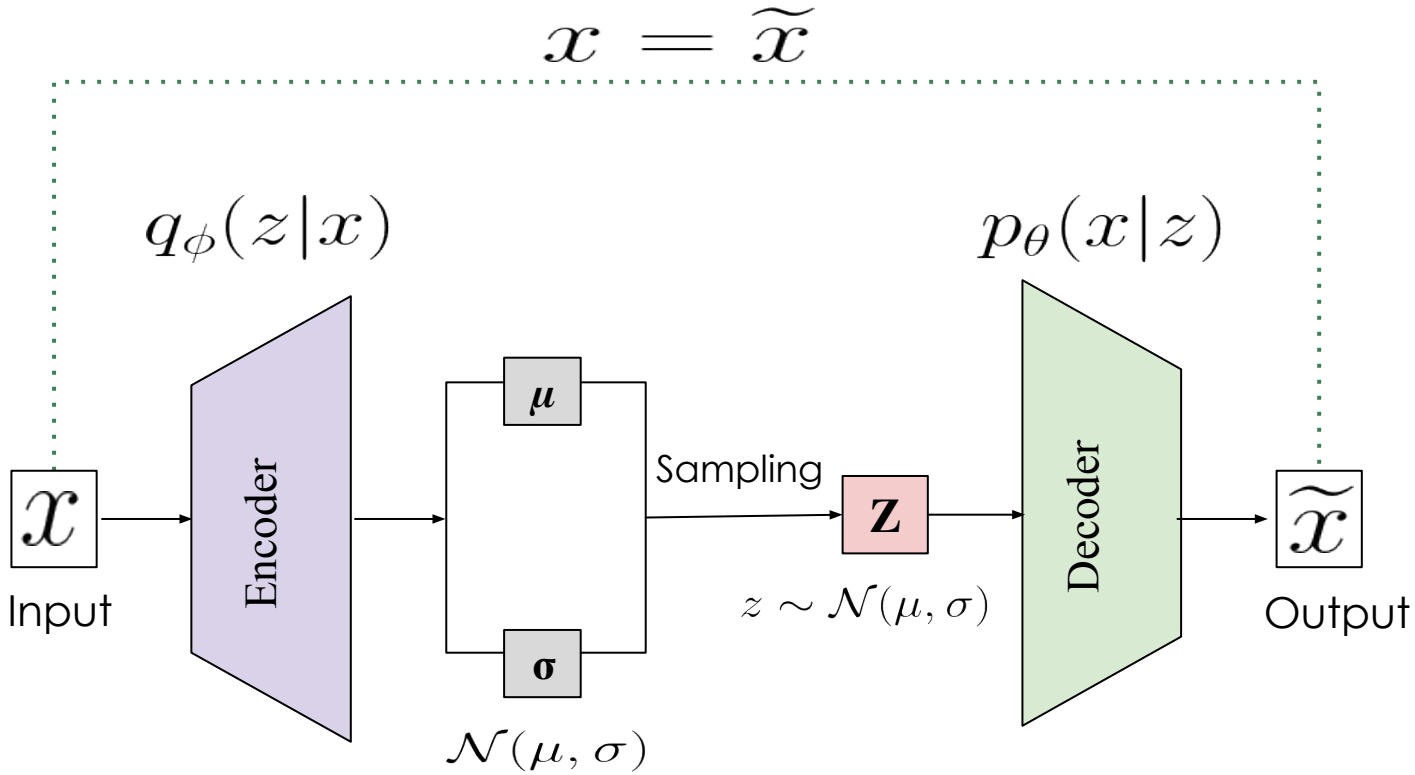
- Consists of two Neural Networks (NNs):
  - Encoder: learns how to reduce the input space and compress the input data into a lower representation.
  - Decoder: Learns how to reconstruct the data from the encoded representation to be close to the original input.
  - Reconstruction Loss: Measures the difference between the input and the reconstructed.




$$L(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left[ x_i - decoder_\theta(encoder_\phi(x_i)) \right]$$

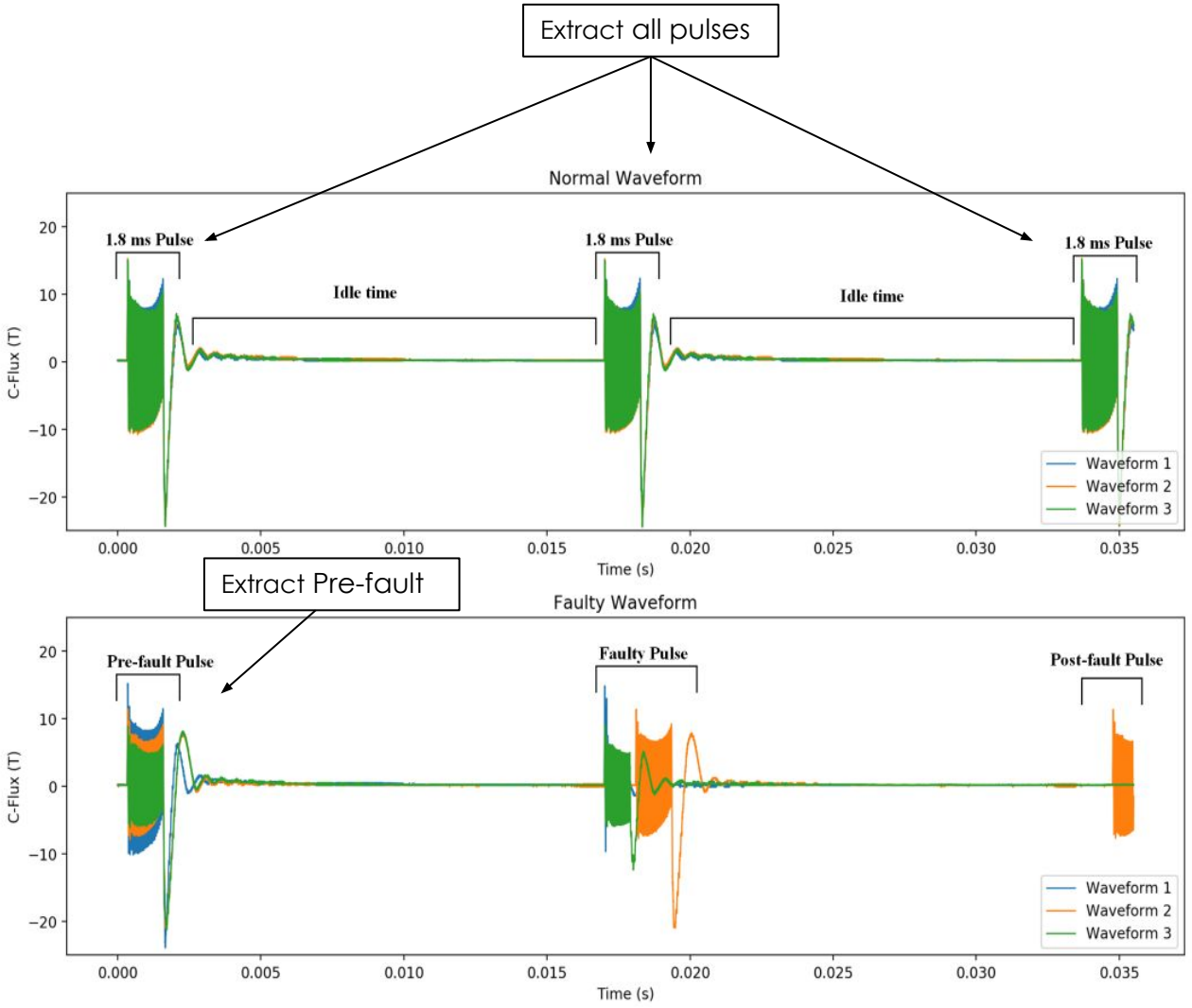
# What is Variational Autoencoder (VAE)

- AE learns a function to map each input from the training data to a number and then learns the reverse mapping.
- VAE learns probability distribution of the input by estimating a mean and standard deviation of the that distribution.
- Then, use the estimated parameters (mean and std) to reconstruct the input data.
- VAE consists of:
  - Encoder: outputs parameters of a pre-defined distribution in the latent space.
  - Sampling layer: takes the estimated mean and std to sample a distribution.
  - Decoder: Learns how to reconstruct the data from the learned distributions.
- The loss function consists of:
  - Kullback–Leibler divergence (prior & estimated)
  - Reconstruction error (input & output)



# Data Preparation

- Normal file: we extract all three micro-pulses and label it as "Normal".
  - Each 1.8 ms macro-pulse has 4500 time steps (i.e. sampling rate is 400 ns).
- Abnormal file: we extract the first micro-pulse (pre-fault) and label it as "Abnormal".
  - Only extract the pre-fault to allow detecting the anomalies ahead of time.
  - Each 1.8 ms macro-pulse has 4500 time steps (i.e. sampling rate is 400 ns).
- We do this for 14 different waveforms:
  - Three magnetic fluxes: A-FLUX, B-FLUX, C-FLUX.
  - Six IGBT current waveforms: A+, A+*, B+, B+*, C+, and C+*.
  - Two waveforms that represent the cab bank voltage.
  - Two waveforms represent the modulator output voltage.
  - One waveform represents the time change (i.e. dV/dt).




*Courtesy to Majdi Radaideh, ORNL

# Normal Data Descriptions

- We have four main modules: SCL, CCL, DTL, and RFQ with a total of 7246 waveforms distributed as in Fig. 2 and each module has sub-modules as in Fig 3.

- After data preparation, the waveforms are saved in a 3D tensor of shape (*samples, time, features*).
  - Samples: depend on the module we choose
  - Time: 4500 time-steps of each macro-pulse
  - Features: the 14 waveforms saved by the controller

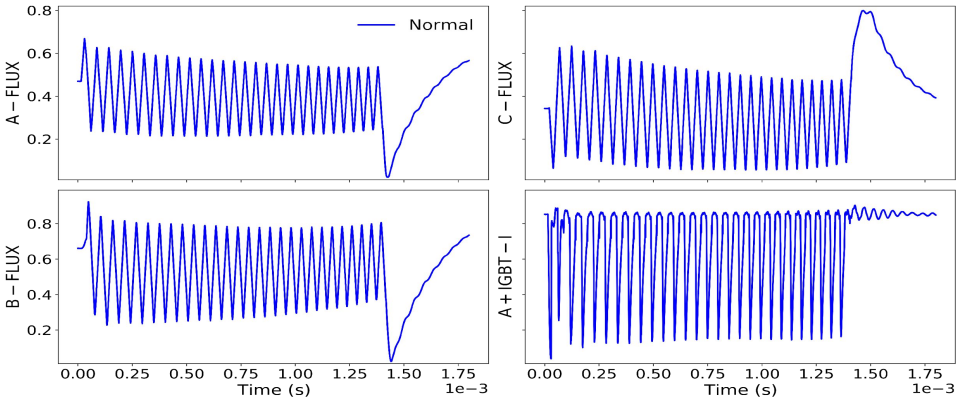



*Figure 4. Examples of 4 different waveforms extracted from SCL module (normalized between 0 and 1).*

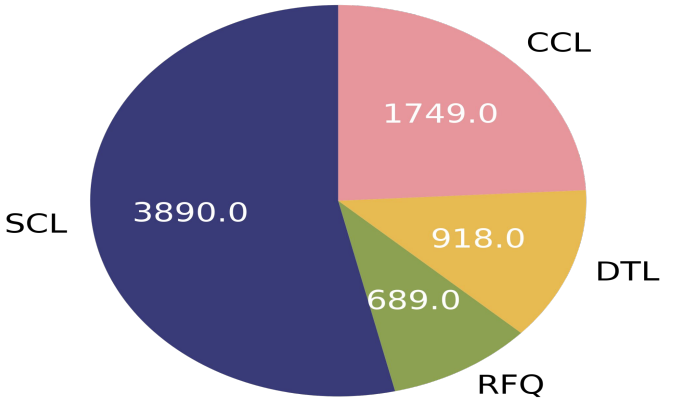



*Figure 2. Number of normal samples for the 4 main modules*

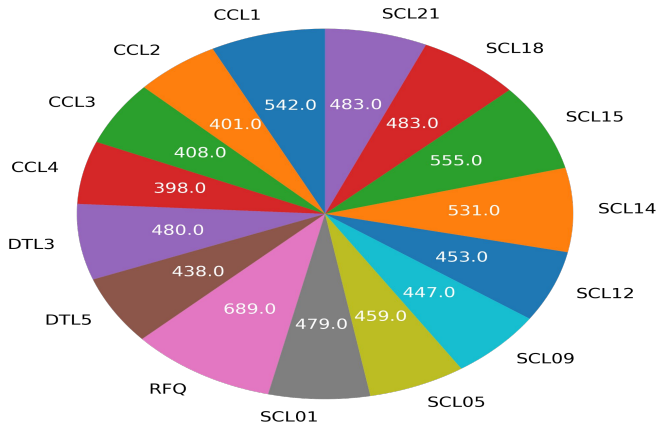



*Figure 3. Number of normal samples for each sub module*

# Abnormal Data Descriptions

- There are several fault types across the modules and there are more than 40 different types (Fig. 5).
- We group fault types into 10 relative categories to increase the number of statistics.
- The grouped faults can be seen in Fig. 8, which represents all faults in all modules.

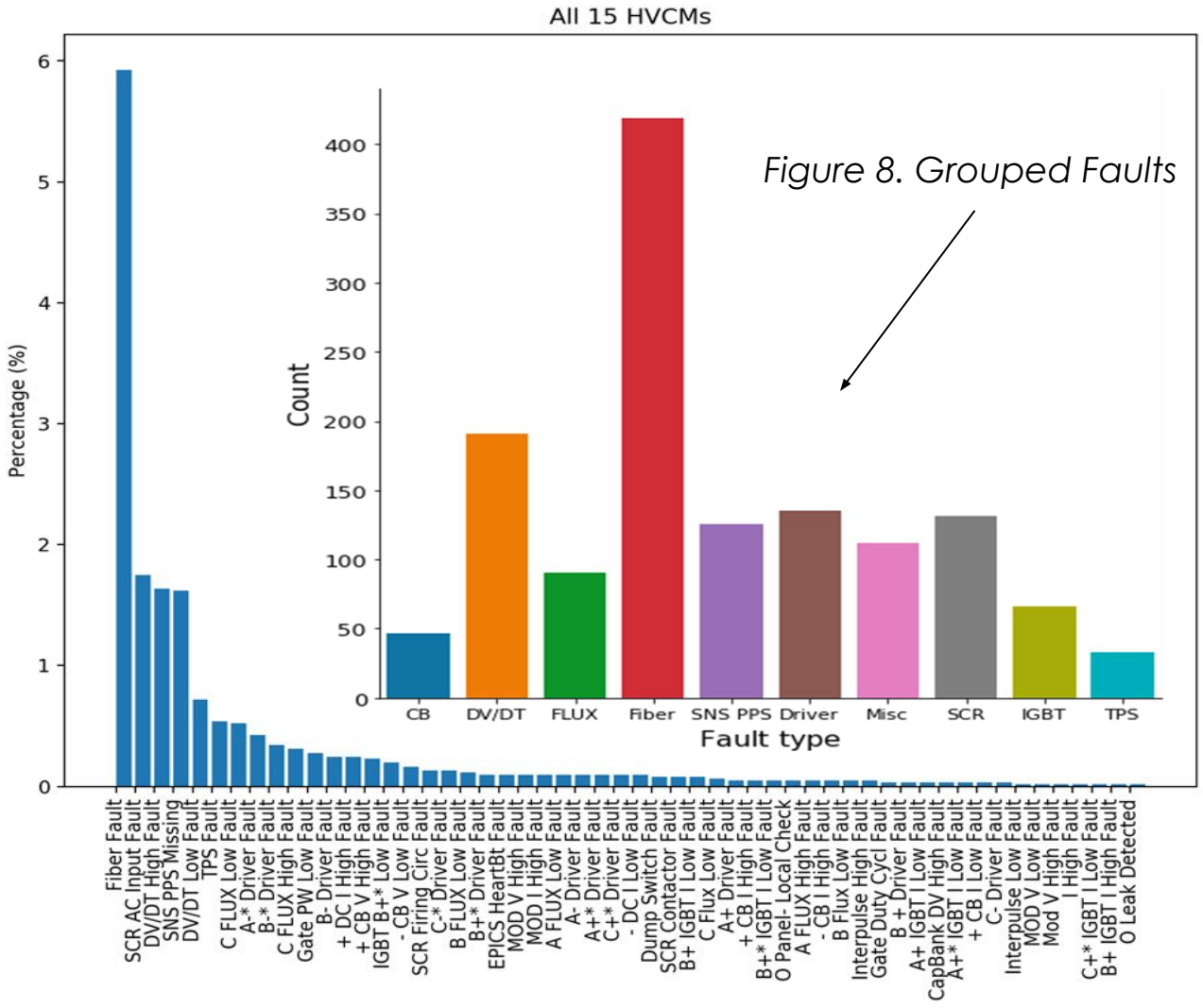



*Figure 8. Grouped Faults*

*Figure 5. Percentage of abnormal waveforms of the overall data*

# Abnormal Data Descriptions

- Some abnormal waveforms can be identified using clustering or simple visualizations as shown in the box plot in Fig 6.
- However, many other examples fall within the statistics of normal data and cannot be easily separated, so we need a more accurate tool to detect anomalies.

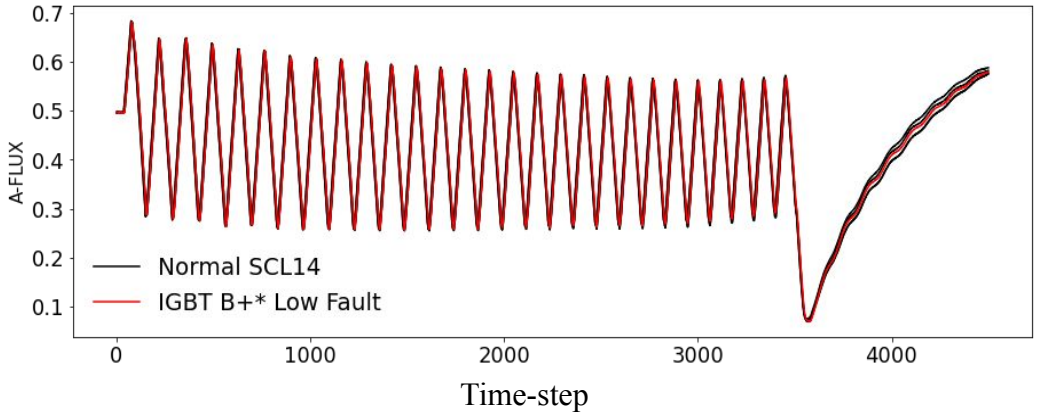



*Figure 7. Two faulty waveforms. Left can be identified easily, while we need reconstruction error to detect the fault in the right example.*

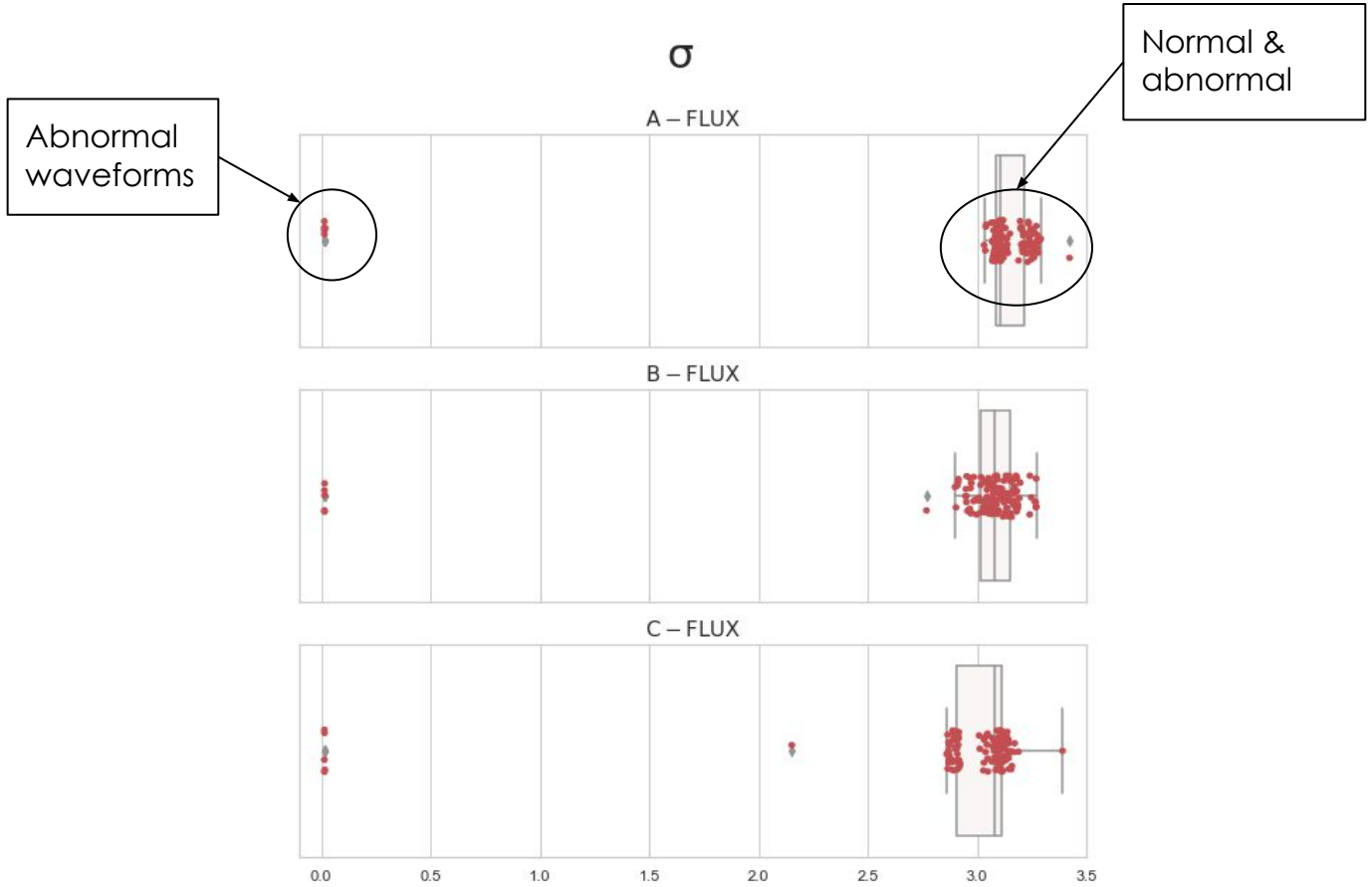



*Figure 6. The std distributions of normal & abnormal waveforms.*

# Single Module-based VAE

- We adopt the architecture of VAE and use the methodology on several modules trained individually.
- **Goal:** Detect anomalies ahead of time.
- **How**: Train the using "Single Module-based VAE" model on normal waveforms, then at test time we feed abnormal data.
- If the reconstruction error is higher than a threshold value, then we identify the waveform as abnormal.

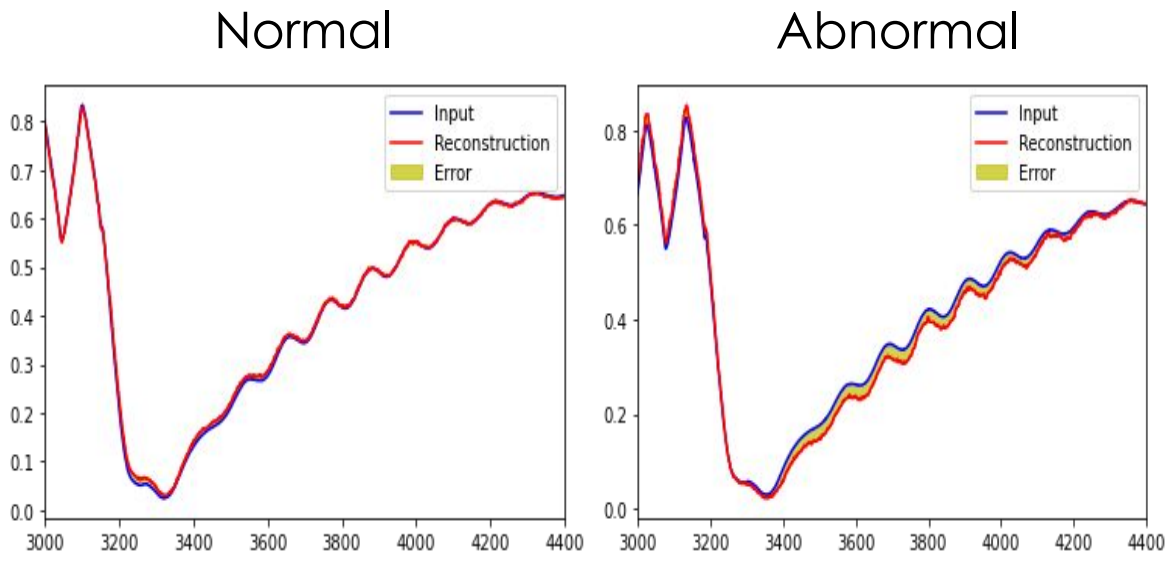
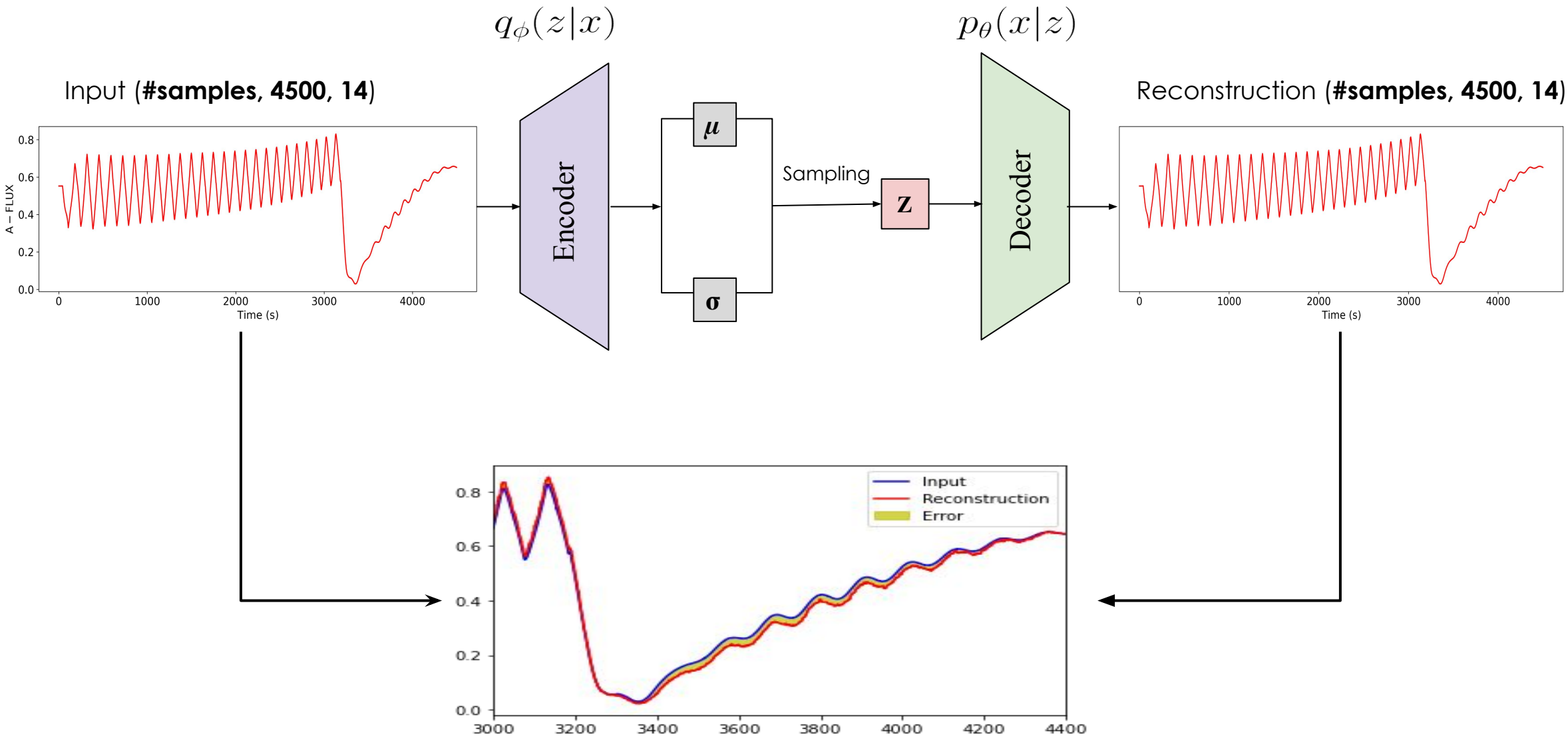



*Figure 8. Left is normal waveform; right is abnormal waveform. The yellow band represents the reconstruction error from our model.*

# Single Module-based VAE Architecture



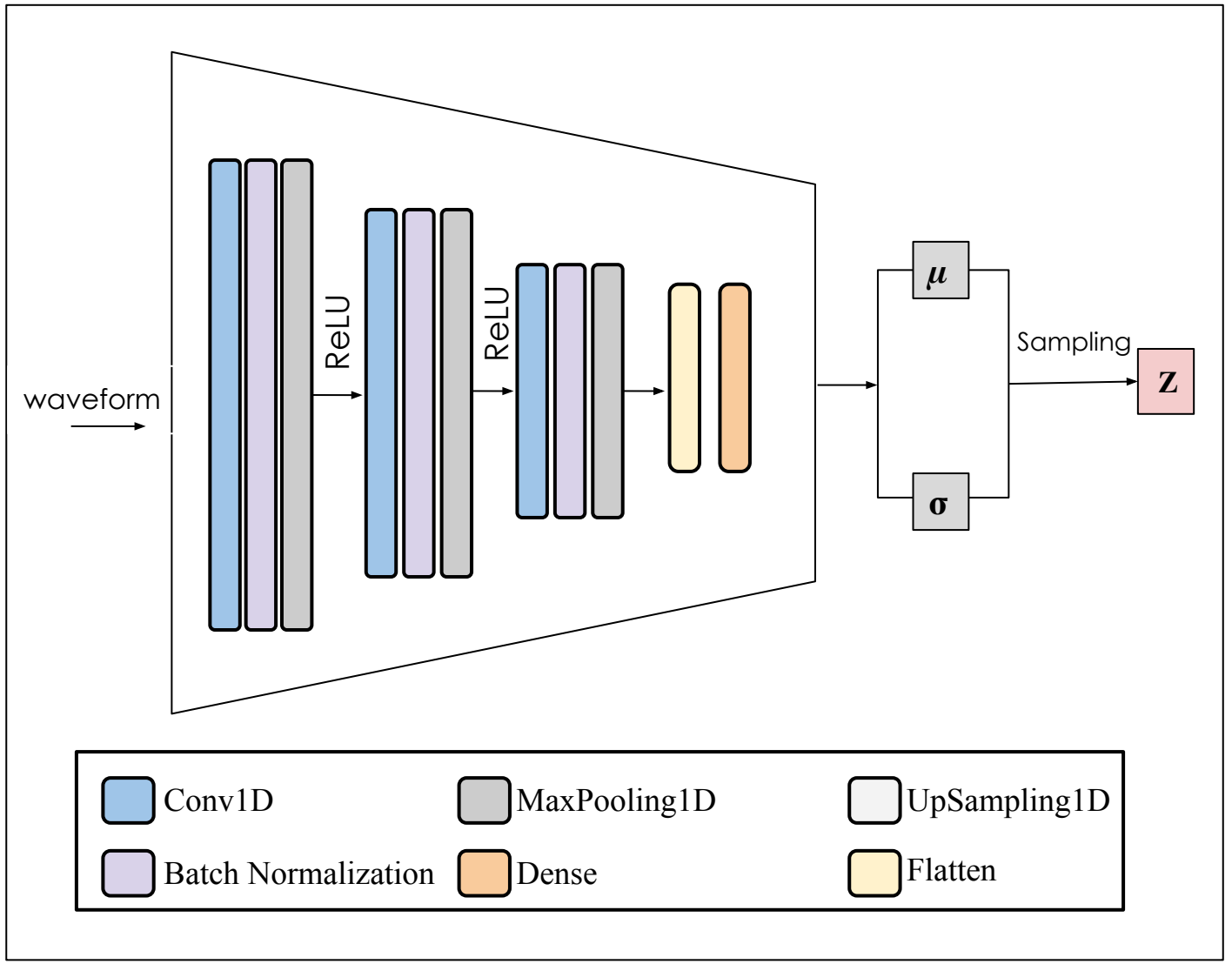



OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

Jefferson Lab

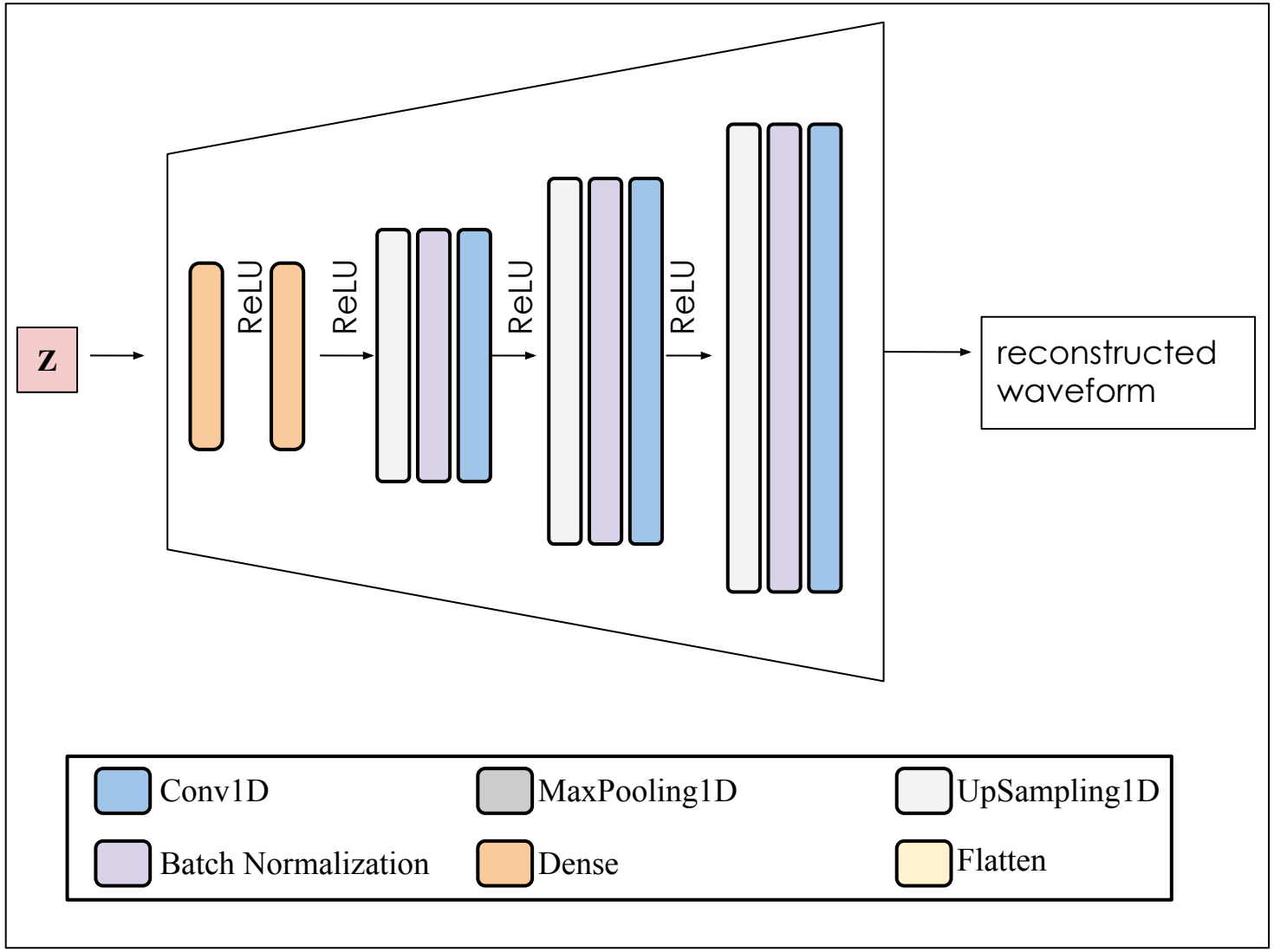
# Encoder Architecture

- Encoder consists of three 1DCNN blocks: Conv1D, BN, 1D pooling.
- Each block is activated by a rectified linear unit (ReLU).
- The output of the CNN blocks is flattened and passed to a fully connected layer (Dense).
- The Dense layer will generate the mean and std of the posterior Gaussian distribution.
- The sampling layer is the final layer of the encoder that is passed to the decoder.



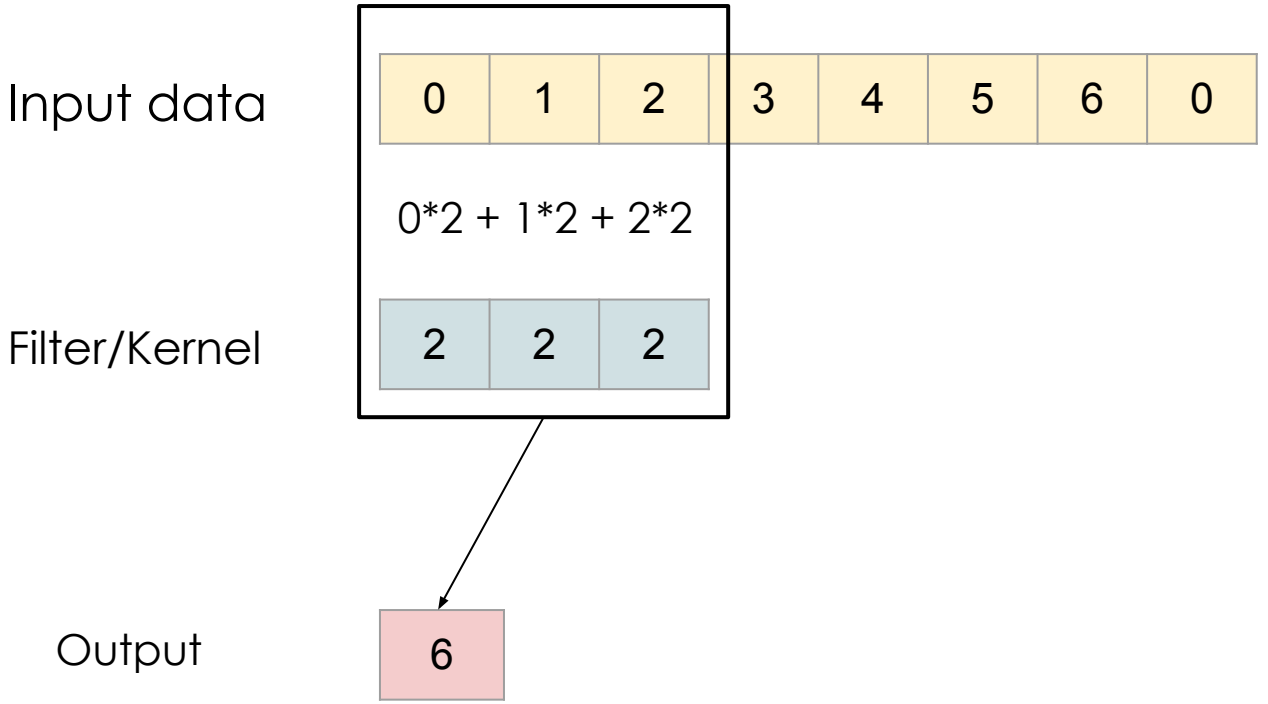
# Decoder Architecture

- Decoder takes z as an input and is passed to Dense layers.
- The output of the Dense layers is reshaped to be fed to the CNN layer.
- Each CNN block consists of: UpSampling, BN, Conv1D followed by a ReLU activation function.
- The output of the last CNN block is the reconstructed inputs.
- The choice of hyper parameters will be discussed later!



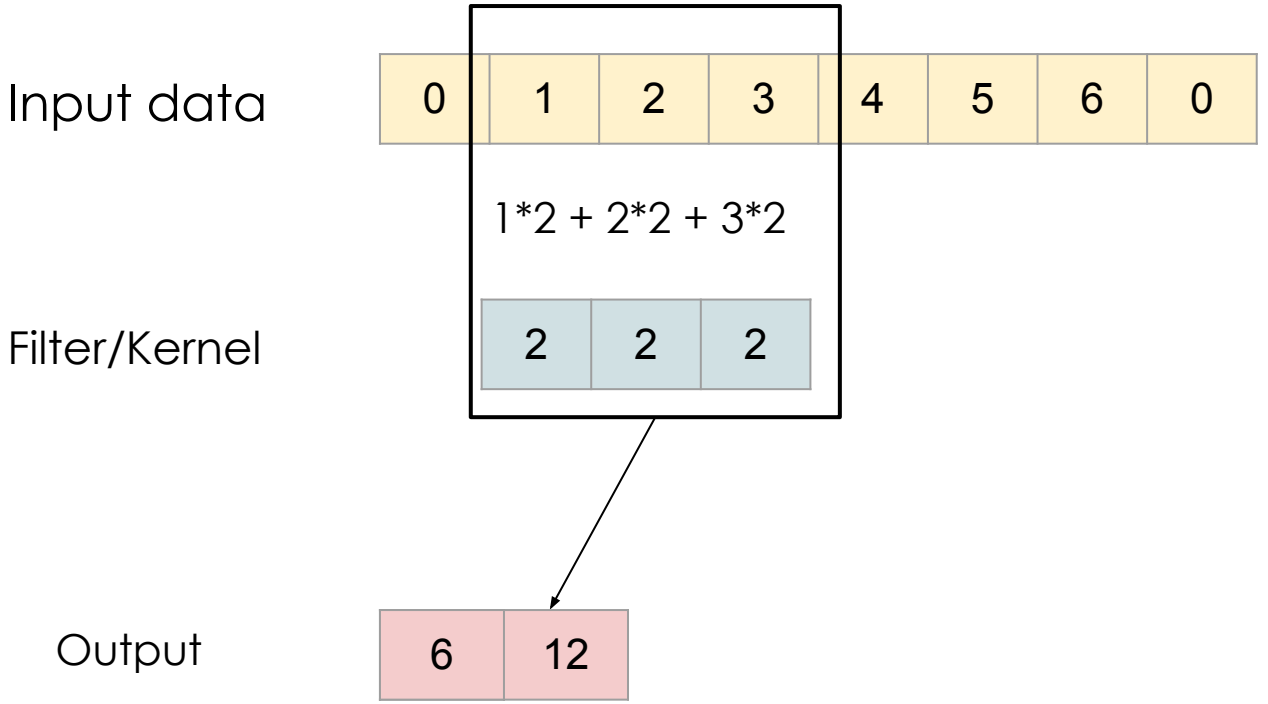
# What is 1D Convolutional Neural Network (1DCNN)

- It is a layer consists of convolutional operations using filers/kernels.
- Filters are initialized randomly and updated for each iteration.
- Filter size and the stride (shift) are hyper-parameters that often require domain knowledge.
- After initializing the filters, we perform dot product with the input vector.
- Then move by the stride value and do the conventional operations.



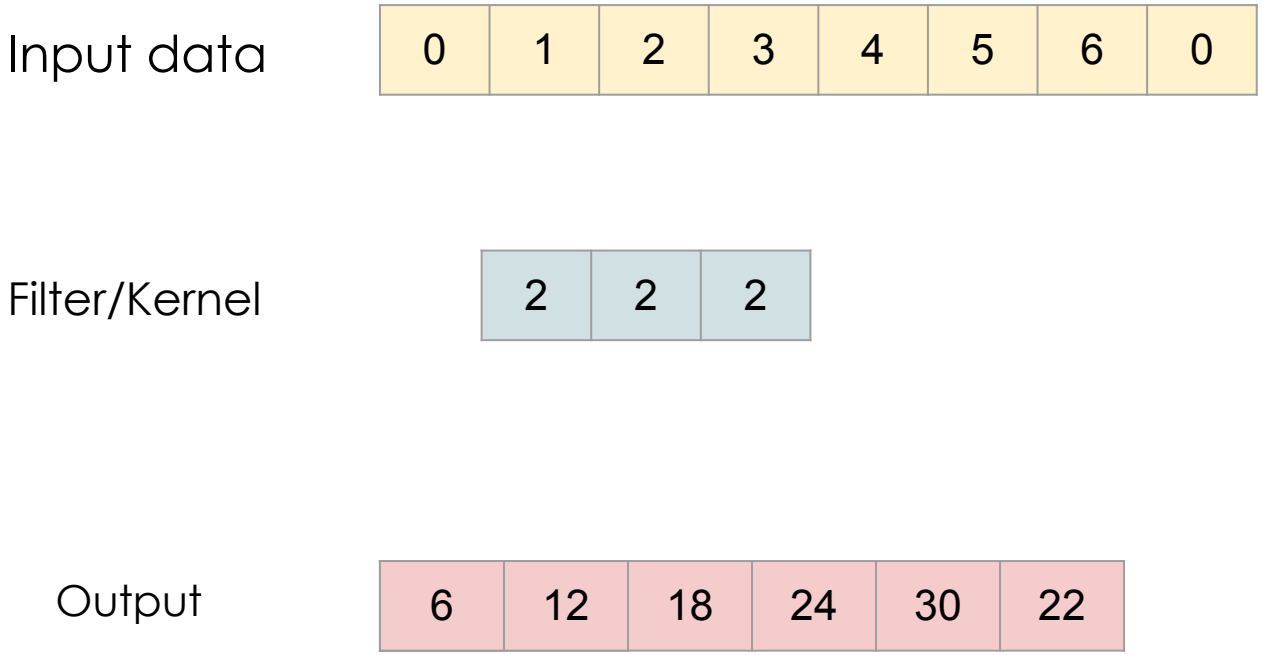
# What is 1D Convolutional Neural Network (1DCNN)

- It is a layer consists of convolutional operations using filers/kernels.
- Filters are initialized randomly and updated for each iteration.
- Filter size and the stride (shift) are hyper-parameters that often require domain knowledge.
- After initializing the filters, we perform dot product with the input vector.
- Then move by the stride value and do the conventional operations.



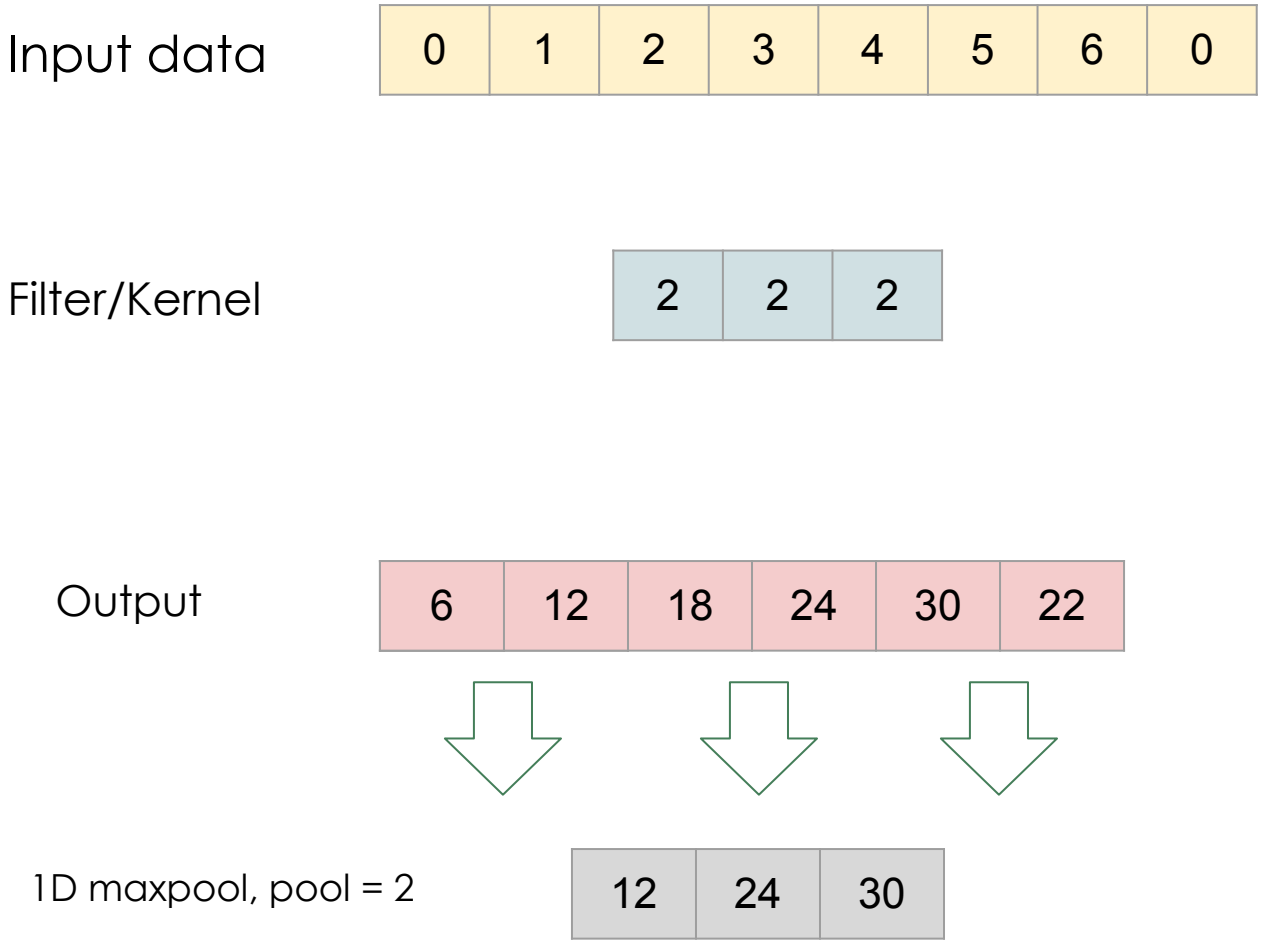
# What is 1D Convolutional Neural Network (1DCNN)

- It is a layer consists of convolutional operations using filers/kernels.
- Filters are initialized randomly and updated for each iteration.
- Filter size and the stride (shift) are hyper-parameters that often require domain knowledge.
- After initializing the filters, we perform dot product with the input vector.
- Then move by the stride value and do the conventional operations.



# What is 1D Convolutional Neural Network (1DCNN)

- It is a layer consists of convolutional operations using filers/kernels.
- Filters are initialized randomly and updated for each iteration.
- Filter size and the stride (shift) are hyper-parameters that often require domain knowledge.
- After initializing the filters, we perform dot product with the input vector.
- Then move by the stride value and do the conventional operations.



# Single Module-based Results

- Training: We train the model on SCL5 module with 80% as training and leave 20% as a test set.
- Testing: We test the model accuracy to reconstruct normal waveforms, and how the reconstruction error vary between normal and abnormal data.

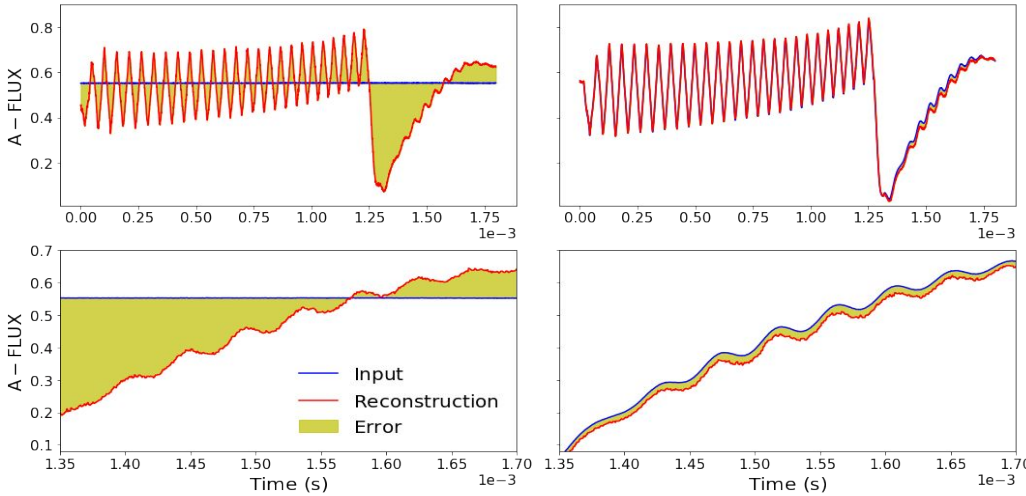



*Figure 10. Reconstruction of two abnormal waveforms.*

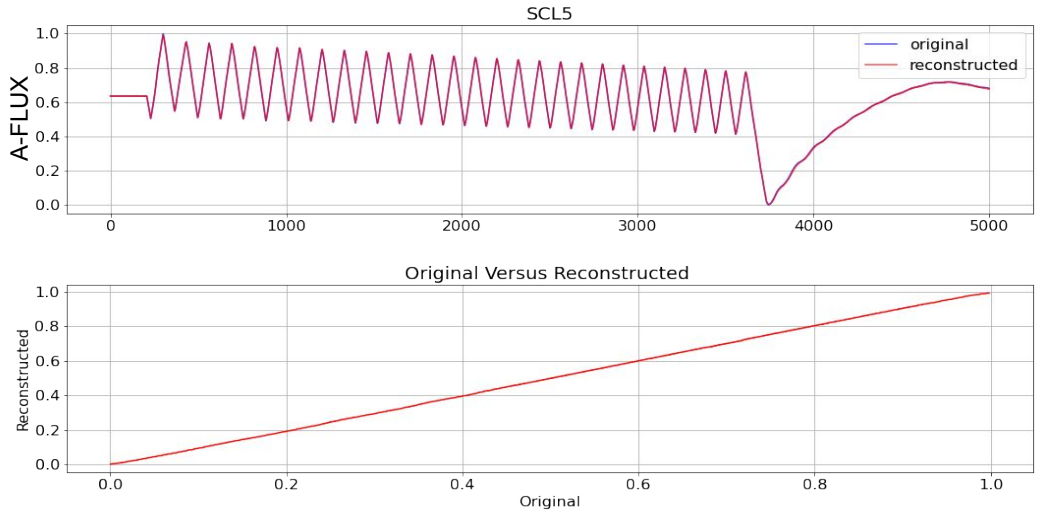



*Figure 8. Reconstruction of normal data (test set).*

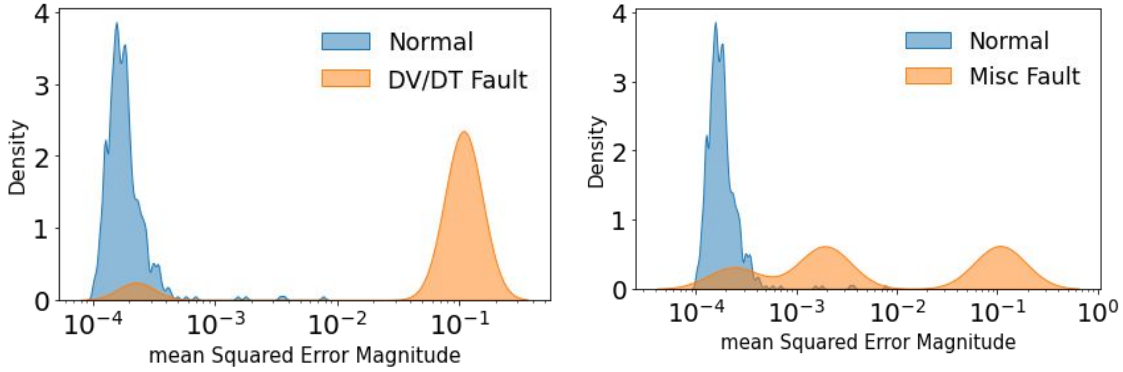



*Figure 9. Density plot when reconstructing normal And abnormal waveforms for two fault types.*

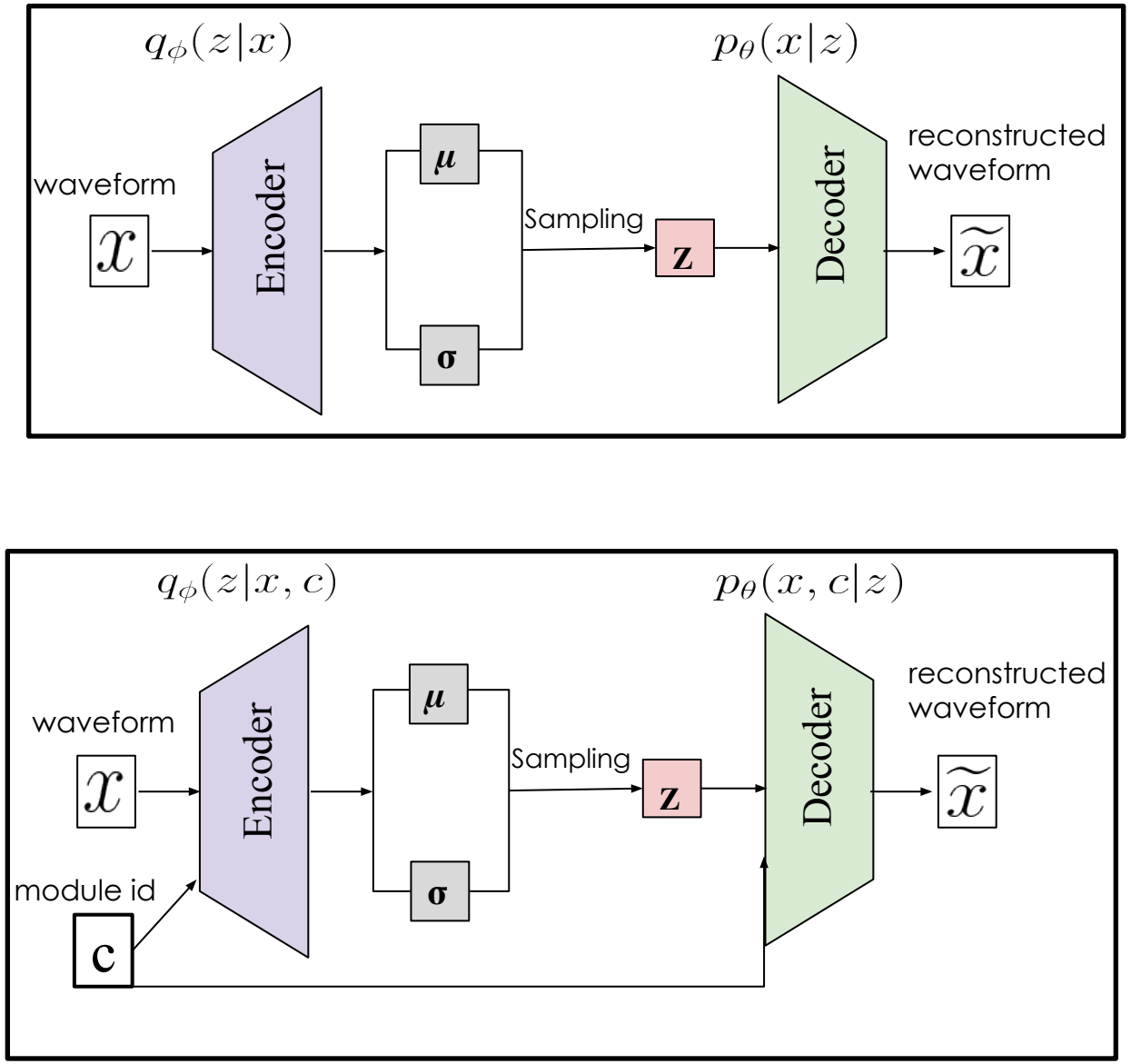


OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

Jefferson Lab

# Extension to Multi Module-based

- Instead of training the model on all 15 modules individually, we extend our methodology to multiple modules.
- We use one framework to train all 15 modules together.
  - Increase the number of statistics.
  - Improve the generalization of the model.
- We are motivated by the idea of Conditional Variational Autoencoder (CVAE)
  - Condition the model on labels to associate each file with its label
  - This will associate each waveform with its module.
- It can learn several normal waveforms across all modules which increases the sensitivity of detecting anomalies.



# Multi Module-based VAE Results

- Training: We train the model on all 15 modules module with 80% as training and leave 20% as a test set.
- Testing: We test the model accuracy to reconstruct inputs as well as the model ability to distinguish between normal and abnormal.
- Since the model was trained on 14 waveforms, we also investigate which waveform is more important.
- We can see in the density plot using Mod-V we have more separation between normal waveforms and DV/DT fault.
- The Receiver Operating Curve (ROC): shows the performance measurement for the classification at various threshold values.
- The Area Under the ROC (AUC): measures the entire area and shows the degree of separability, where 1 is the ideal case.

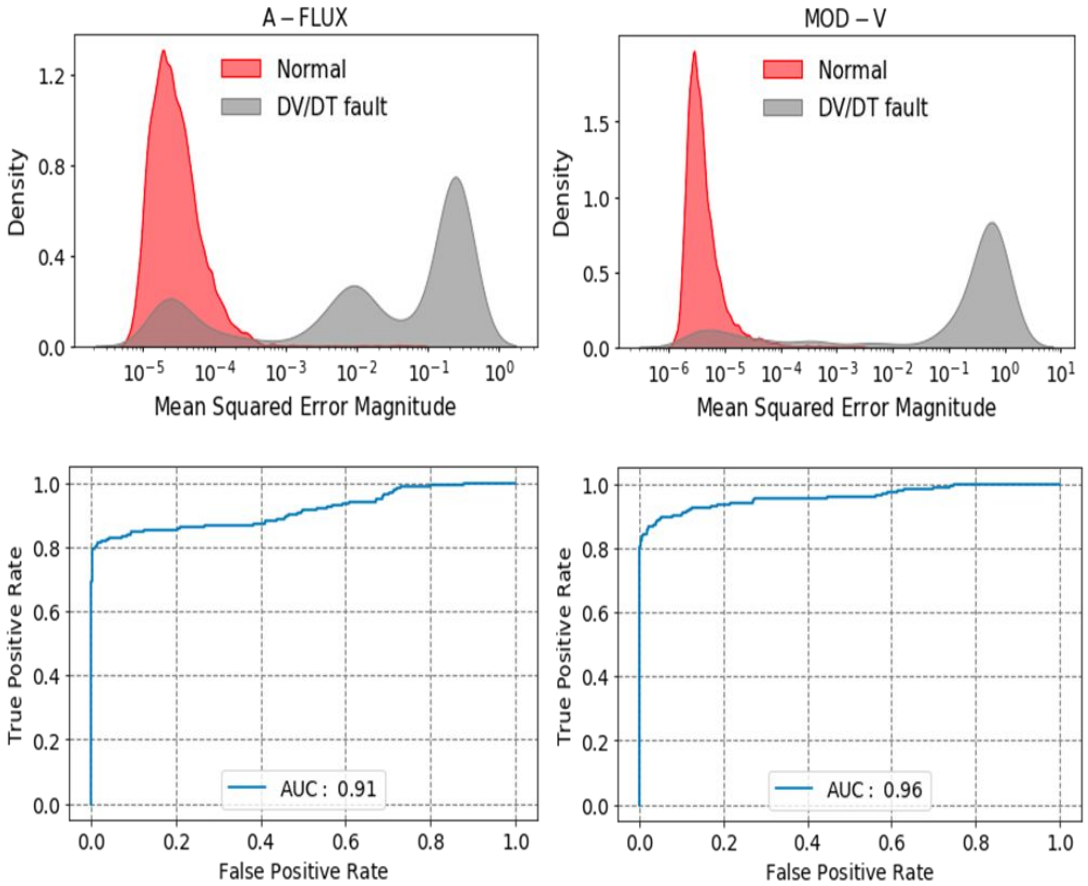



*Figure 11. Density plot when reconstructing normal And abnormal waveforms for two fault types.*

# Single Module vs Multi Module Comparison

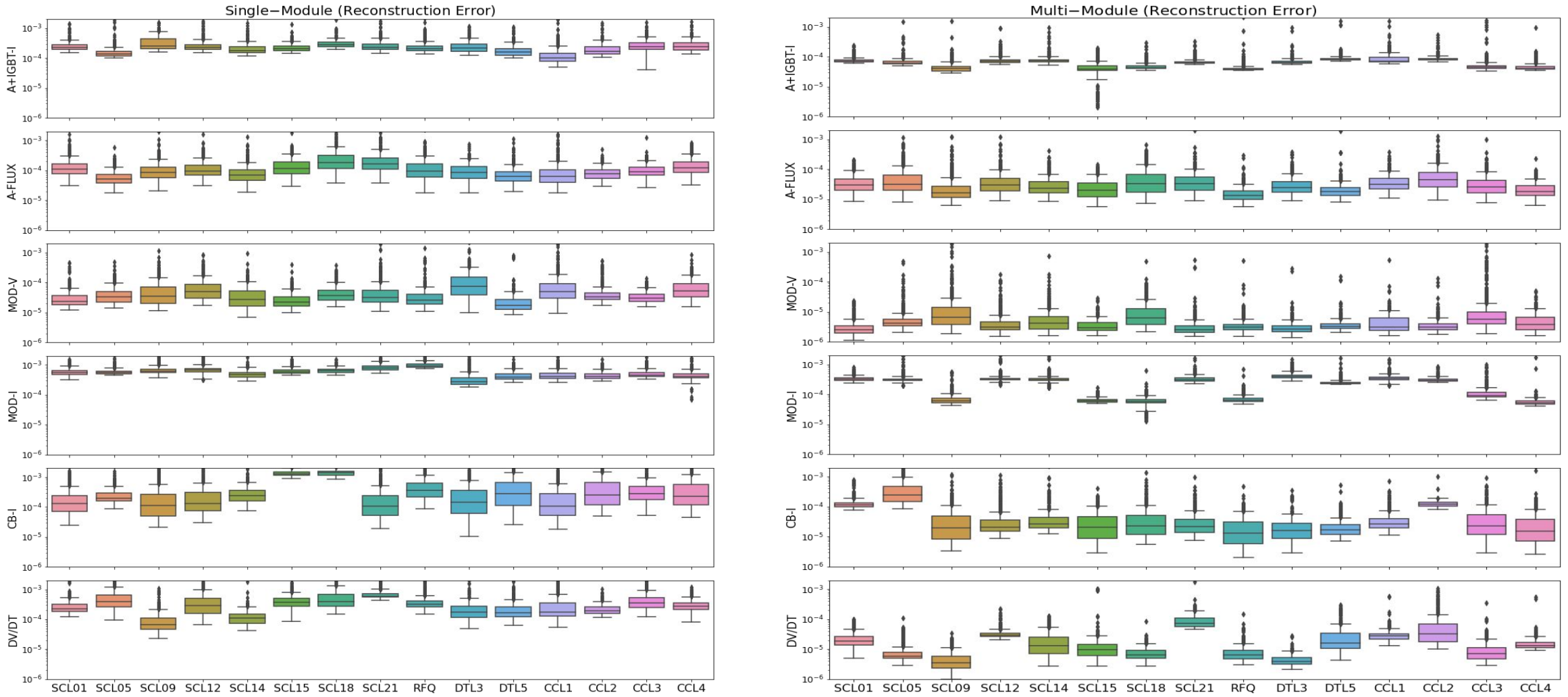



*Figure 12. Box plot shows the reconstruction error when feeding normal waveforms for single module vs multi modules. We can see Multi-Module have overall lower reconstruction error for all scenarios.*

# Outline

- Machine Learning-based Anomaly Detection
- HVCM Anomaly Detection at SNS
  - Introduction
  - Autoencoder & Variational Autoencoder
  - Data preparation
  - Single Module-based model
  - Multi Module-based model
- Model Evaluation
  - Loss Landscape visualization

# Model Evaluation

- The performance of NNs can be impacted by several factors, such as variable initialization, optimizers, network architectures, batch sizes, and other hyper-parameters.
- Studying the effects of various hyper-parameters is challenging because their loss values live in a high-dimensional space.
- Several scientific applications rely on a simple (1D line) loss curve which is computing the mean/sum of the loss for each epoch which produces a scalar, and then plot the loss values as a function of epochs as shown in Fig. 13 and 14.
- While this method is beneficial to give an overview of the model performance, it only shows a small range of gradients of the parameters, and it does not show the convexity of the function, and why certain NNs architectures generalize better than others.
- Visualizing the loss surface has been used recently which can give us a better understanding of the NNs training behavior and can explain the model generalization.

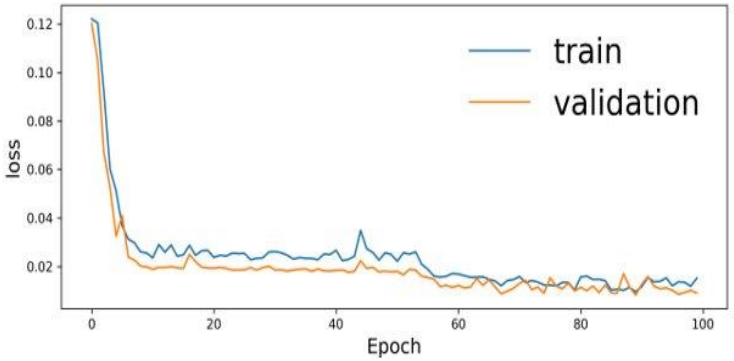



*Figure 13. Loss values of VAE trained on CCL module*

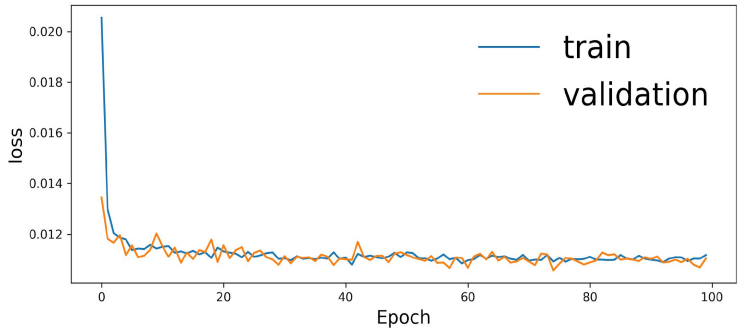
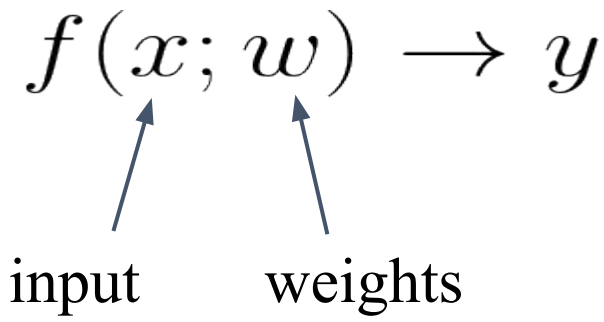



*Figure 14. Loss values of VAE trained on all modules*

# Neural Network Loss Function

- NN can be thought of as a function that maps input x to the output y.
- This function has a set of parameters "weights"
  - In many deep learning applications, there are millions or billions of parameters.



# Neural Network Loss Function

- NN can be thought of as a function that maps input x to the output y.
- This function has a set of parameters "weights"
  - In many deep learning applications, there are millions or billions of parameters.
- We train the NN by minimizing a loss function that penalizes the difference between f(x) and y.

$$f(x; w) \longrightarrow y$$

input weights

$$L(w) = min \sum_i \| f(x_i; w) - y_i \|^2$$

# Neural Network Loss Function

- NN can be thought of as a function that maps input x to the output y.
- This function has a set of parameters "weights"
  - In many deep learning applications, there are millions or billions of parameters.
- We train the NN by minimizing a loss function that penalizes the difference between f(x) and y.
- The loss function is a function of the weight parameters so it has a very high dimensionality.
- In this example, if $f$ is linear, we expect it will be a convex function;

$$f(x; w) \rightarrow y$$

input weights

$$L(w) = min \sum_i \| f(x_i; w) - y_i \|^2$$

# Neural Network Loss Function

- NN can be thought of as a function that maps input x to the output y.
- This function has a set of parameters "weights"
  - In many deep learning applications, there are millions or billions of parameters.
- We train the NN by minimizing a loss function that penalizes the difference between f(x) and y.
- The loss function is a function of the weight parameters, so it has a very high dimensionality.
- In this example, if $f$ is linear, we expect it will be a convex function;
- However, NNs have nonlinearities on top of nonlinearities, so this might be a very non-convex function. (How convex?)

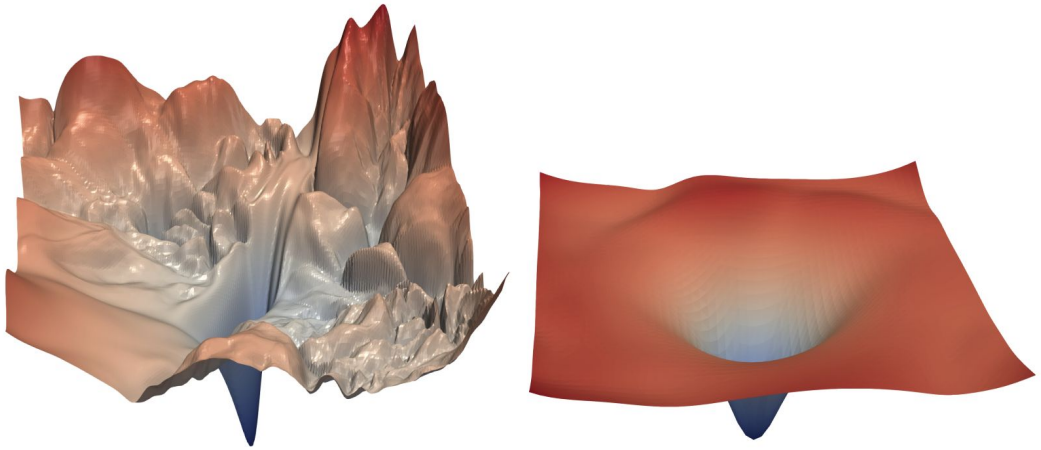
$$f(x; w) \rightarrow y$$

input weights

$$L(w) = min \sum_i \| f(x_i; w) - y_i \|^2$$

# Loss Landscape Visualization

- The loss landscape can show the convexity/non-convexity of the trained models.
- Visualizing the loss landscape can explain why certain choice of NNs architectures are easier to train than others. (i.e. skip connections).
- It can also help in testing the stability and generalization of the trained models.
- In this work, we visualize the loss landscape of our VAE models using Filter Normalization technique and use couture plots.
- We are interested to see the model behavior for the following scenarios:
  - Deep layers V.S. shallow layers
  - Single Module V.S. Multi Module

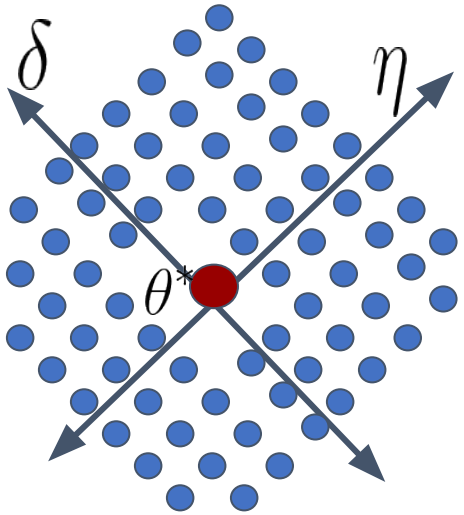


**Hao Li, and et al, Visualizing the Loss Landscape of Neural Nets. NIPS, 2019.*

# Loss Landscape Visualization

- Several methods were proposed to reduce the loss surface into lower dimensional space.
  - Filter Normalization, developed in 2019 by Hao Li and his colleagues.

- Filter normalization steps:
  - Chose a center point $\theta^*$ which represent all the trained weights of our NN.
  - Choose two random directions in weights space $\delta$ and $\eta$.
  - Normalize these two random vectors to have the same norm as the CNN filters.
  - Plot 2D surface of the form: $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$




Visual representation of 2D plane grid with two random directions.

# Loss Landscape: Visualization

- In this example, we train 6 VAE models using different number of CNN layers (L):
  - L = 3, 5, 10, 20, 30, and 40.
- We can see the transition from smooth loss surface to chaotic behavior as the number of layers increased.
- We know in theory deep layers suffer from many problems if no special techniques implemented, now we can see it visually.
- The results give us a hint to fine-tune the model and select the appropriate number of layers and other hyper-parameters.

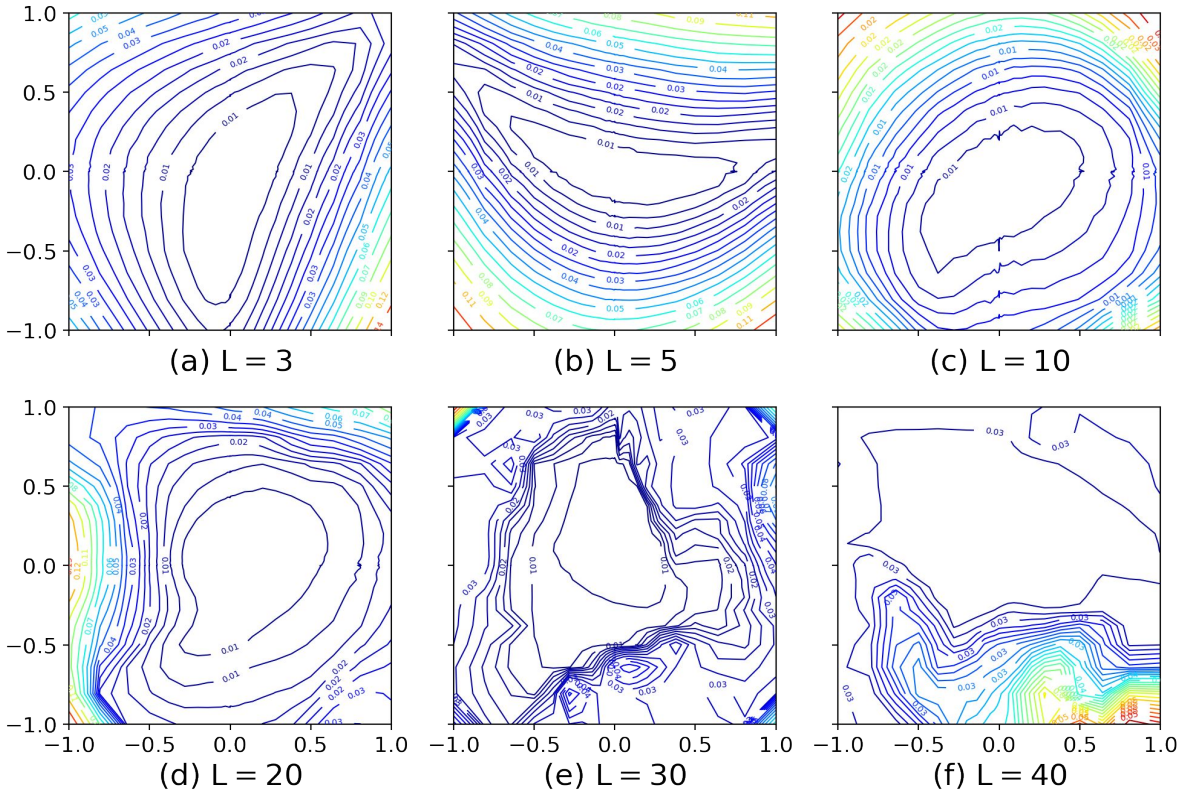
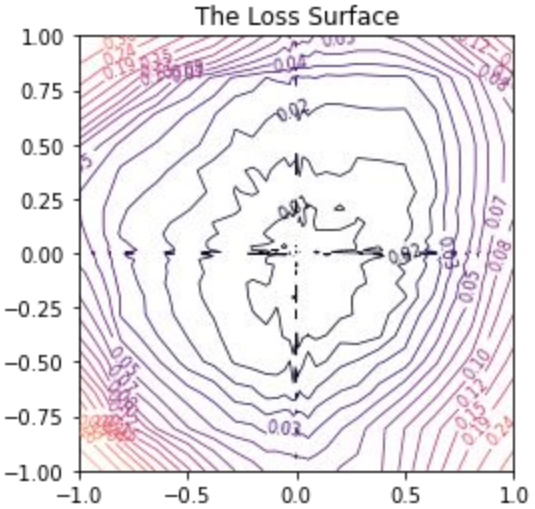



*Figure 11. 2D visualization of the loss surface of the Single Module-based trained using different number of Conv1D layers, where L is the number of layers in the encoder and decoder.*

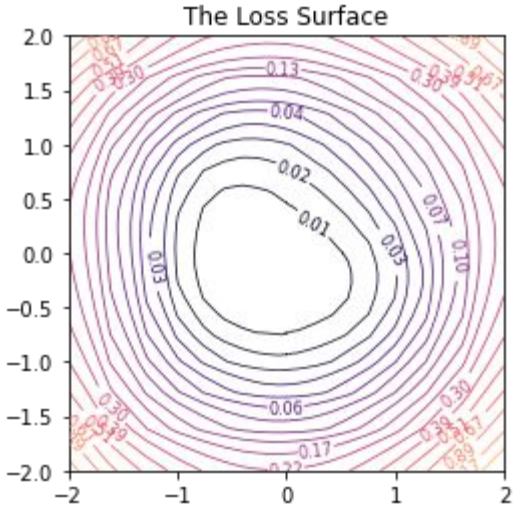
# Loss Landscape: Visualization

- In this example, we train Single Module-based VAE and Multi Module-based VAE.
- Single Module was trained on CCL2, and Multi Module was trained on all 15 modules.
- We train both models with the same number of layers and parameters, and random initializations.
- We can see using the Multi Module-based VAE, we have smother function than the Single Module-based VAE.
- Single Module was also trained on several other modules, we see different behavior where some modules do have smooth loss landscapes.

Single Module-based VAE




Multi Module-based VAE



# Thank you!

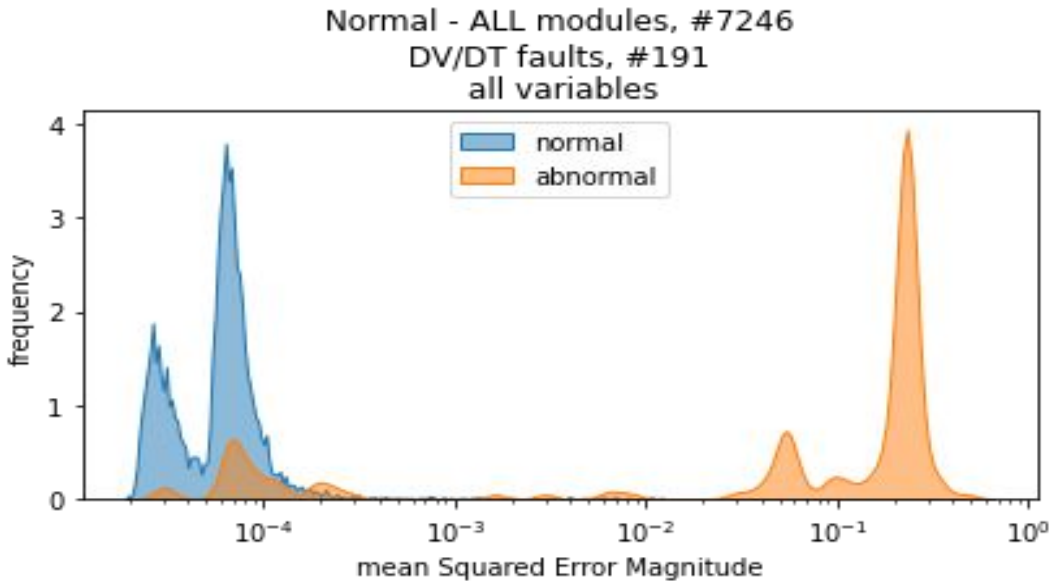
## References:


- Radaideh, Majdi I. and Pappas, Chris and Walden, Jared and Lu, Dan and Vidyaratne, Lasitha and Britton, Thomas and Rajput, Kishansingh and Schram, Malachi and Cousineau, Sarah, Time Series Anomaly Detection in Power Electronics Signals with Recurrent and Convlstm Autoencoders (2022). Available at SSRN: https://ssrn.com/abstract=4069225.

- Kingma, Diederik P and Welling, Max, Auto-Encoding Variational Bayes (2013). Available at https://arxiv.org/abs/1312.6114.

- Li, Hao and Xu, Zheng and Taylor, Gavin and Studer, Christoph and Goldstein, Tom, Visualizing the Loss Landscape of Neural Nets (2017). Available at https://arxiv.org/abs/1712.09913.

# Backup



# HVCM Anomaly Detection at SNS

- The High Voltage Converter Modulators (HVCMs) convert 13.8 kVAC to ~135 kV, 1.35 ms pulses at 60 Hz to the cathodes of klystrons used to accelerate beam at SNS.

- They consist of five major subassemblies:
  - AC switch gear magnetics.
  - A phased controlled rectifier assembly.
  - An air insulated HV enclosure for energy storage capacitors and IGBT switch assemblies
  - An oil insulated HV tank for the high voltage (>2 kV) components
  - A PXI based controller

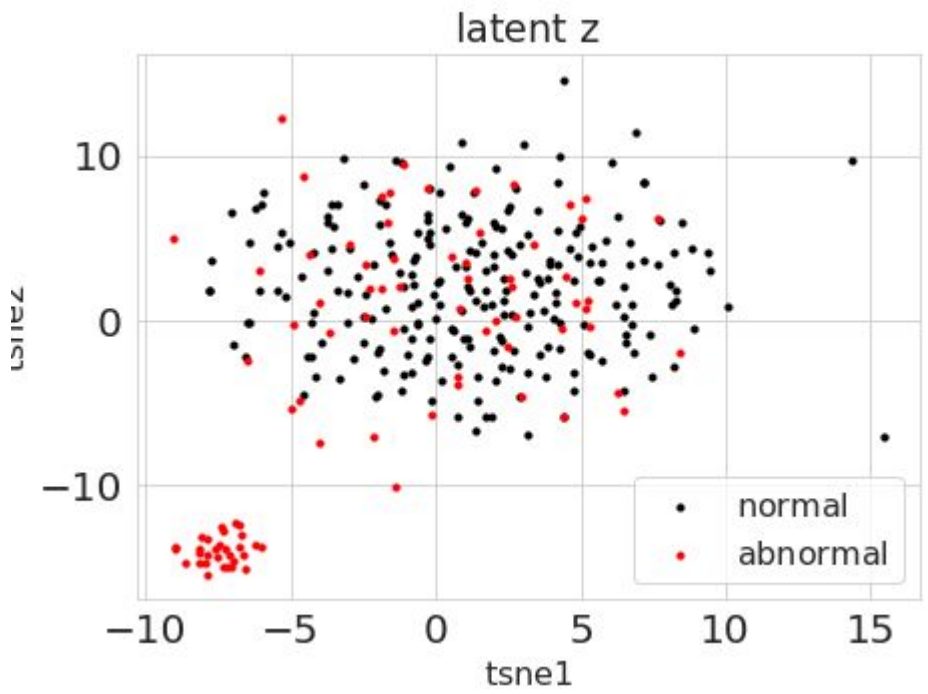
AC magnetics

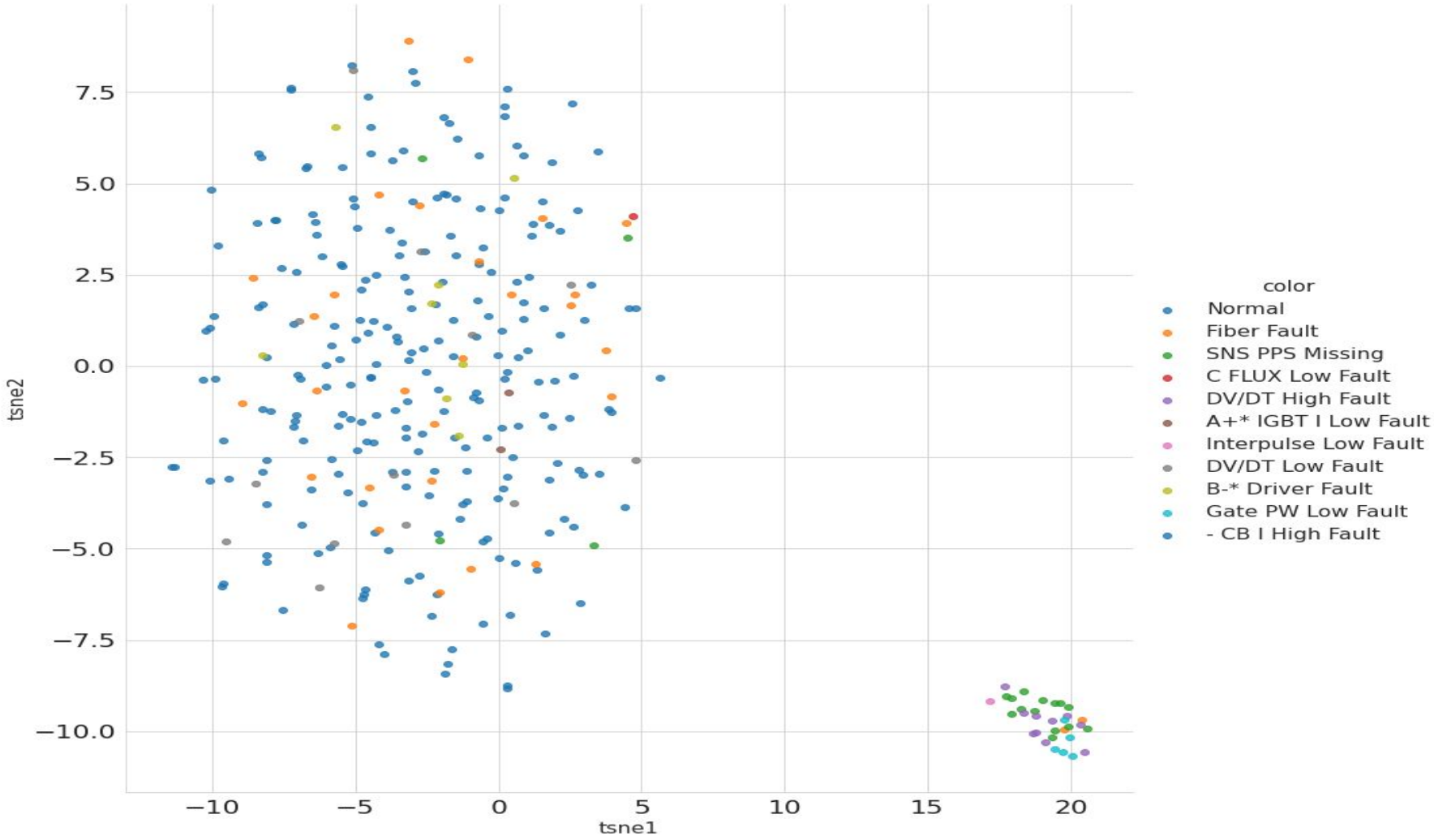
Rectifier

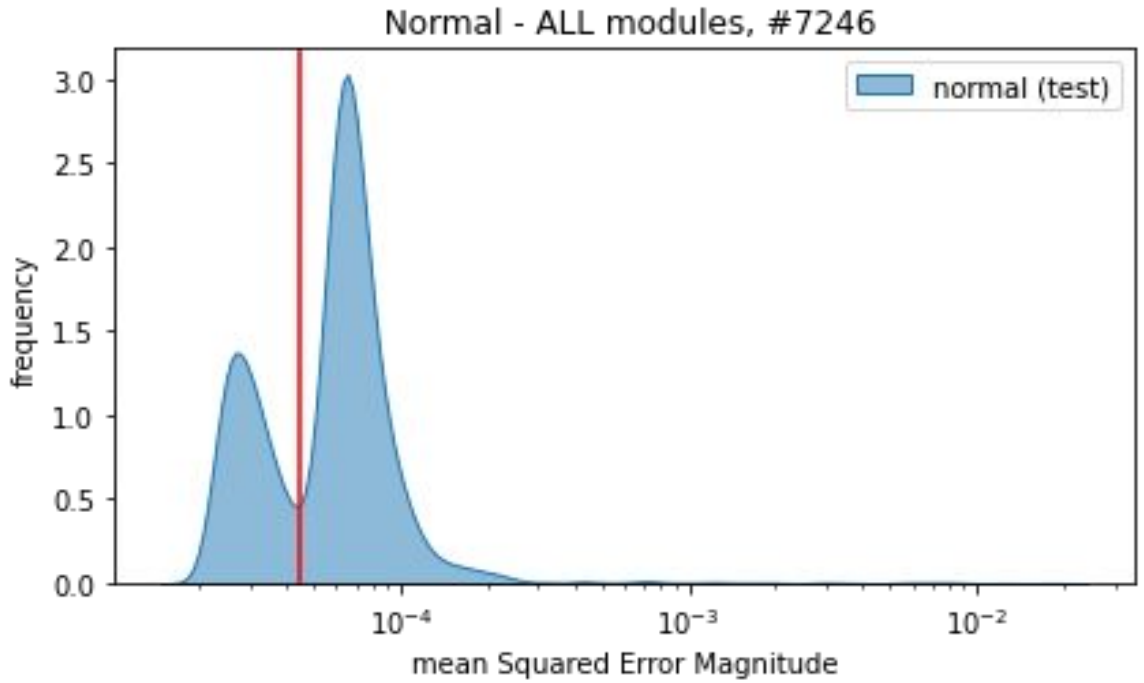
HV enclosure

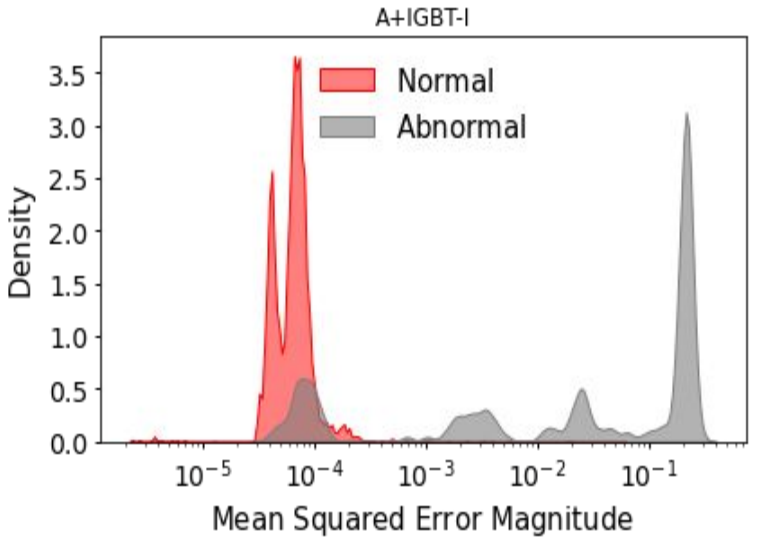
Controls

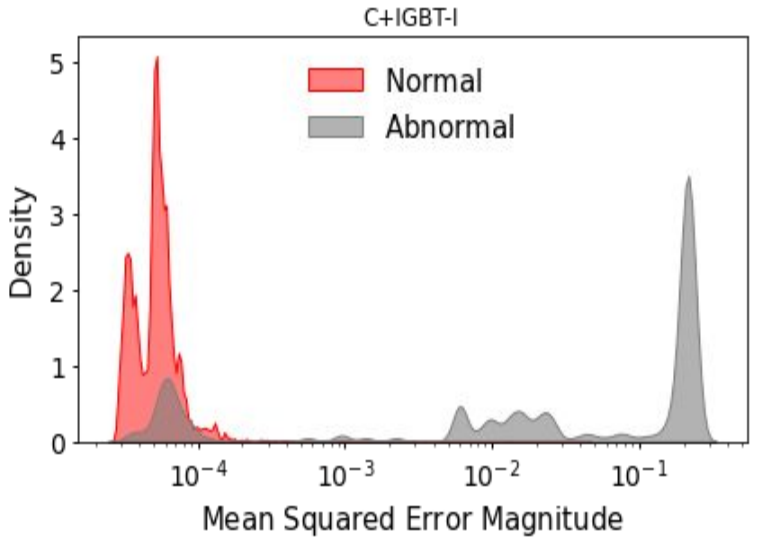

*Courtesy to Chris Pappas

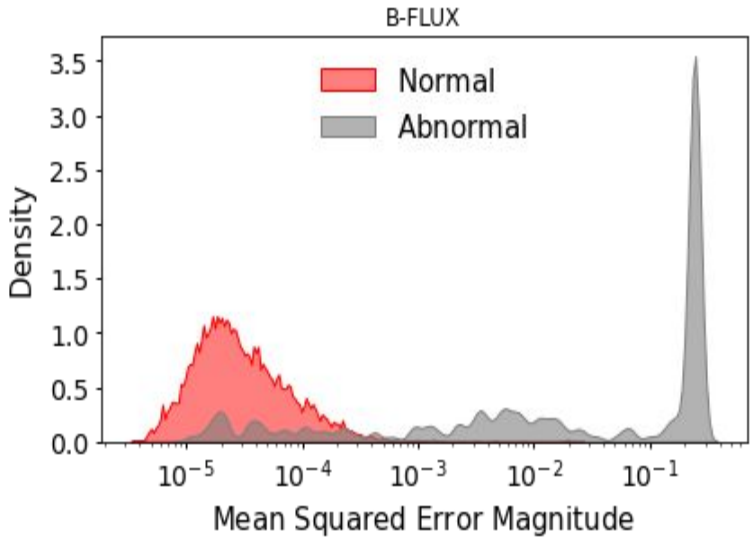


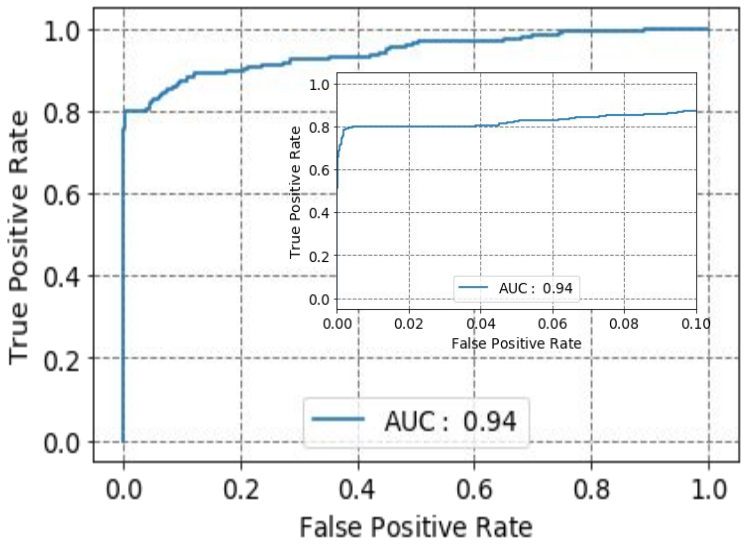

7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
tsne2
−10
−5
0
5
10
15
20
tsne1
color
Normal
Fiber Fault
SNS PPS Missing
C FLUX Low Fault
DV/DT High Fault
A+* IGBT I Low Fault
Interpulse Low Fault
DV/DT Low Fault
B-* Driver Fault
Gate PW Low Fault
- CB I High Fault

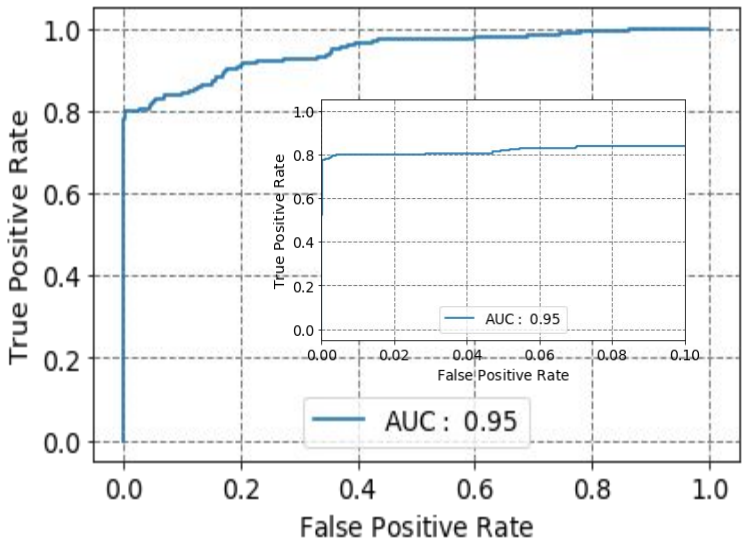


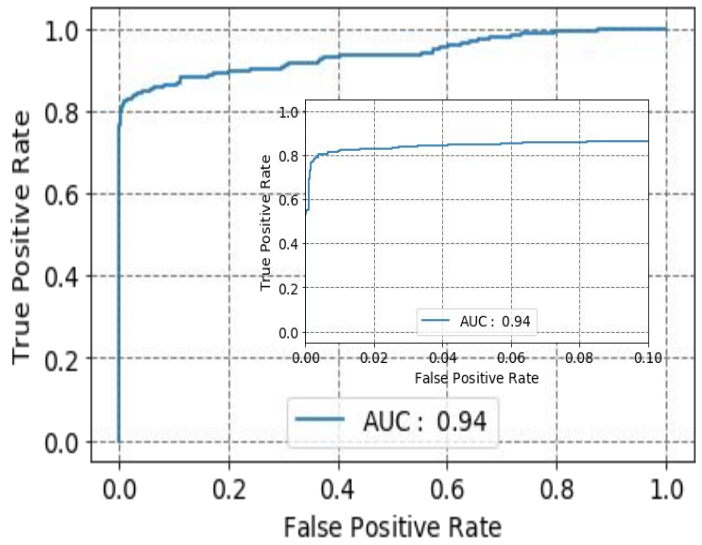

A+IGBT-I
Normal
Abnormal
Density
Mean Squared Error Magnitude

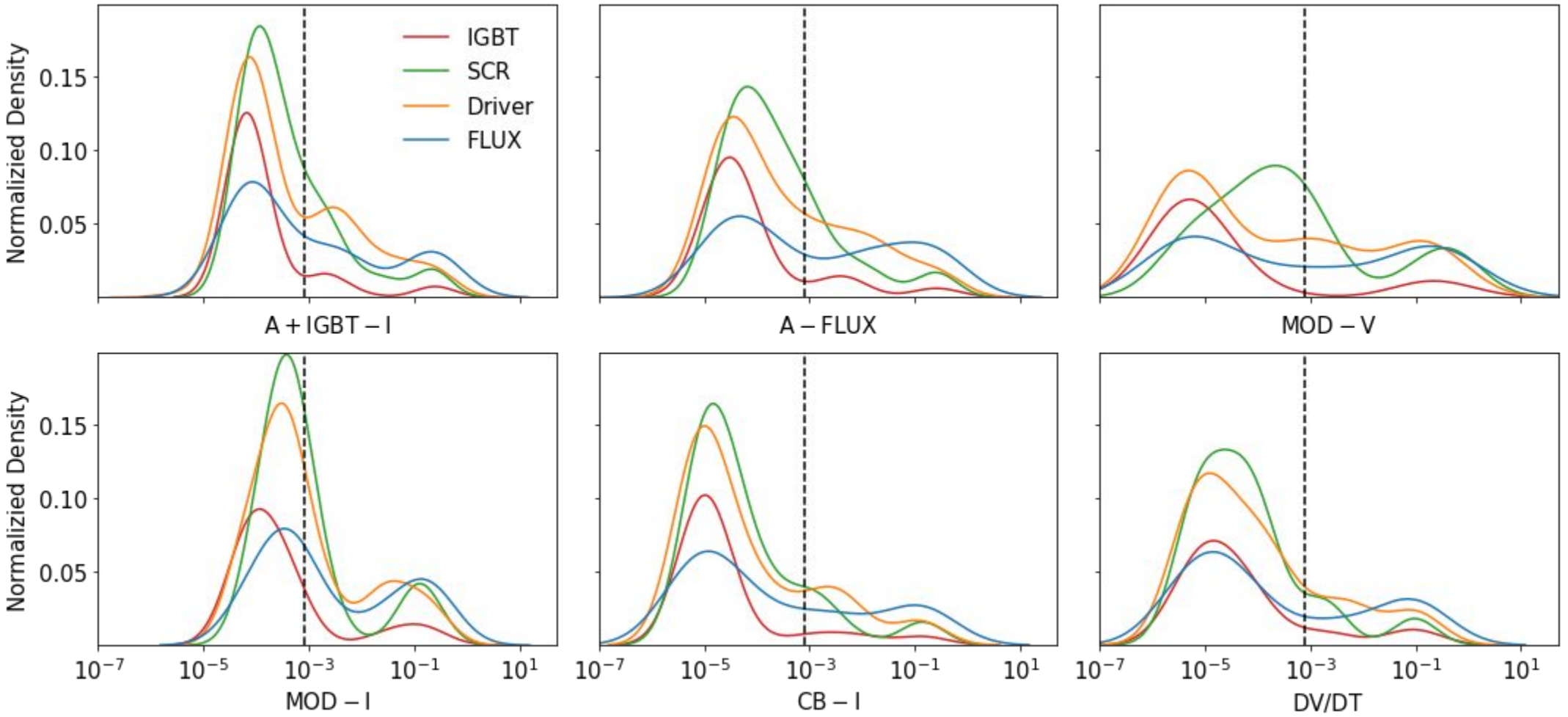

C+IGBT-I
Normal
Abnormal
Density
Mean Squared Error Magnitude

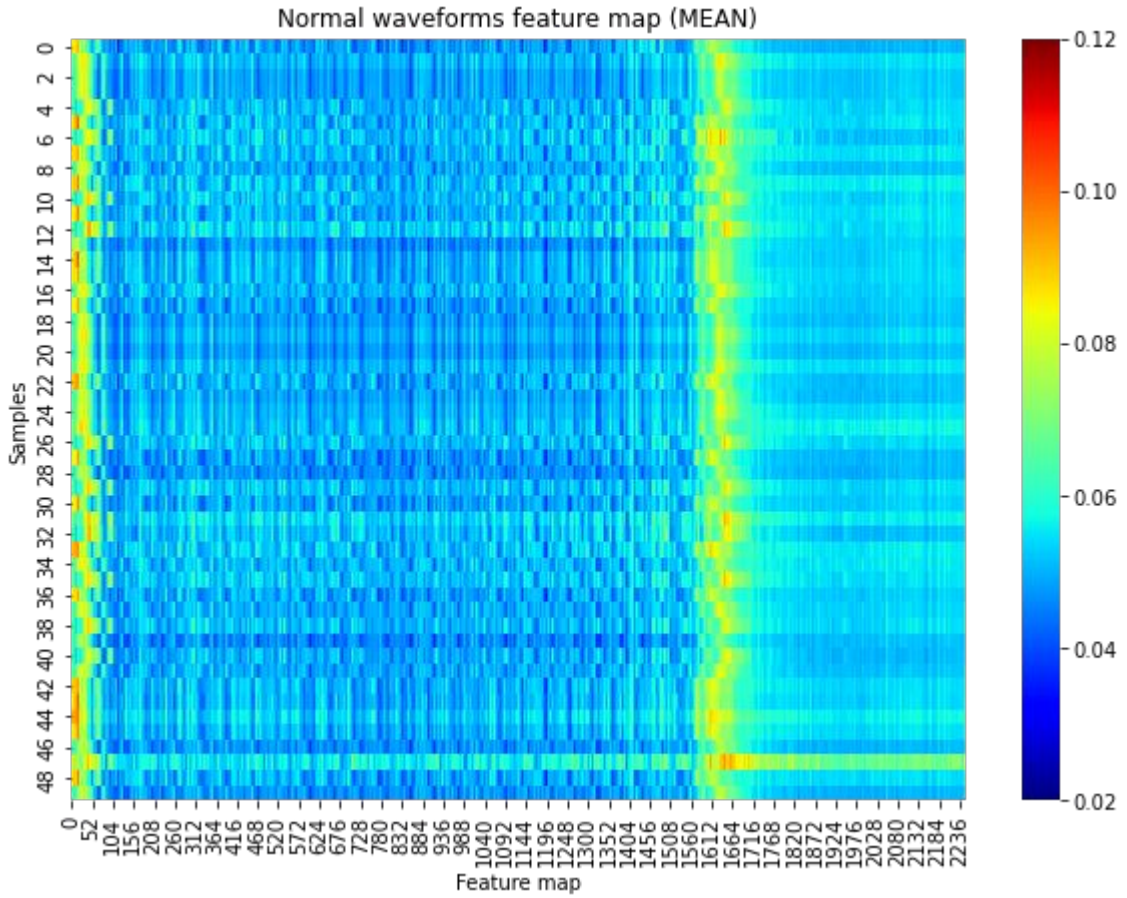

B-FLUX
Normal
Abnormal
Density
Mean Squared Error Magnitude

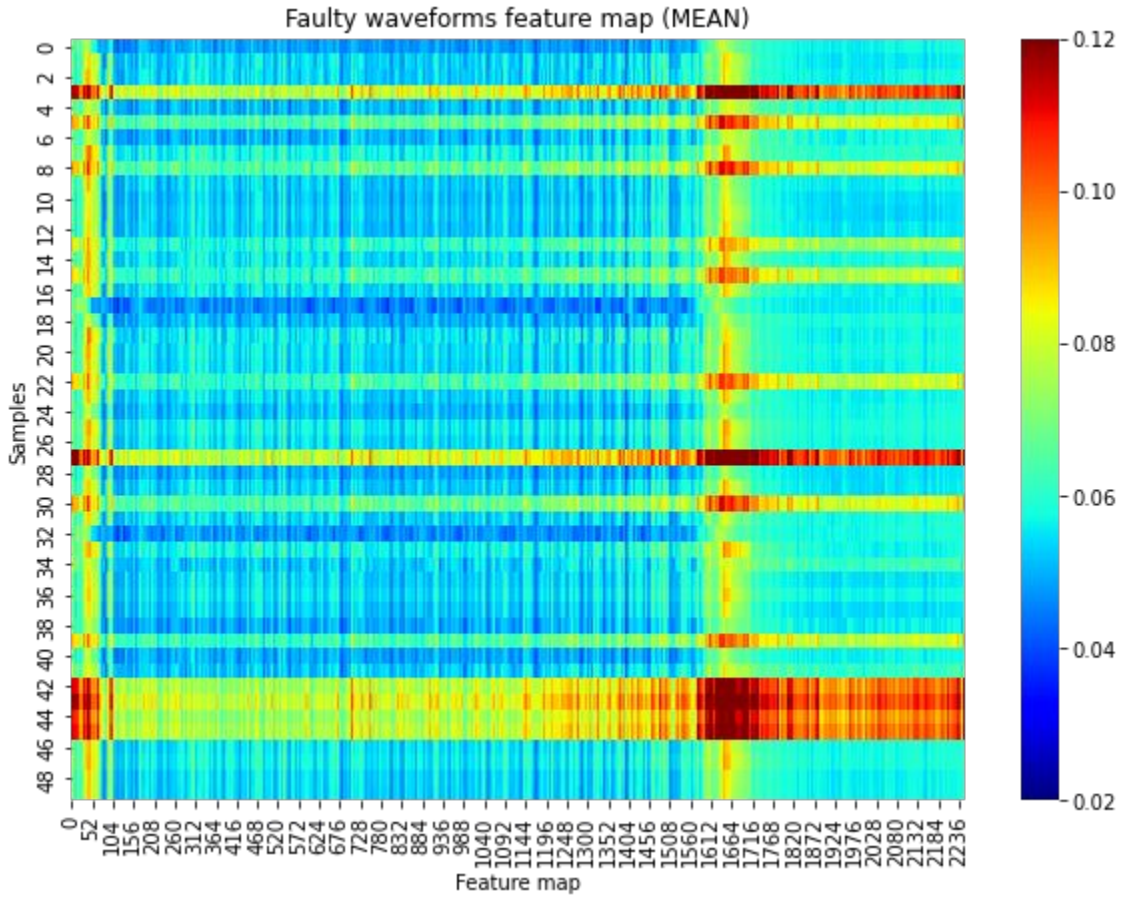

True Positive Rate
False Positive Rate
AUC : 0.94
True Positive Rate
False Positive Rate
AUC : 0.94


True Positive Rate
False Positive Rate
AUC : 0.95
True Positive Rate
False Positive Rate
AUC : 0.95


True Positive Rate
False Positive Rate
AUC : 0.94
True Positive Rate
False Positive Rate
AUC : 0.94

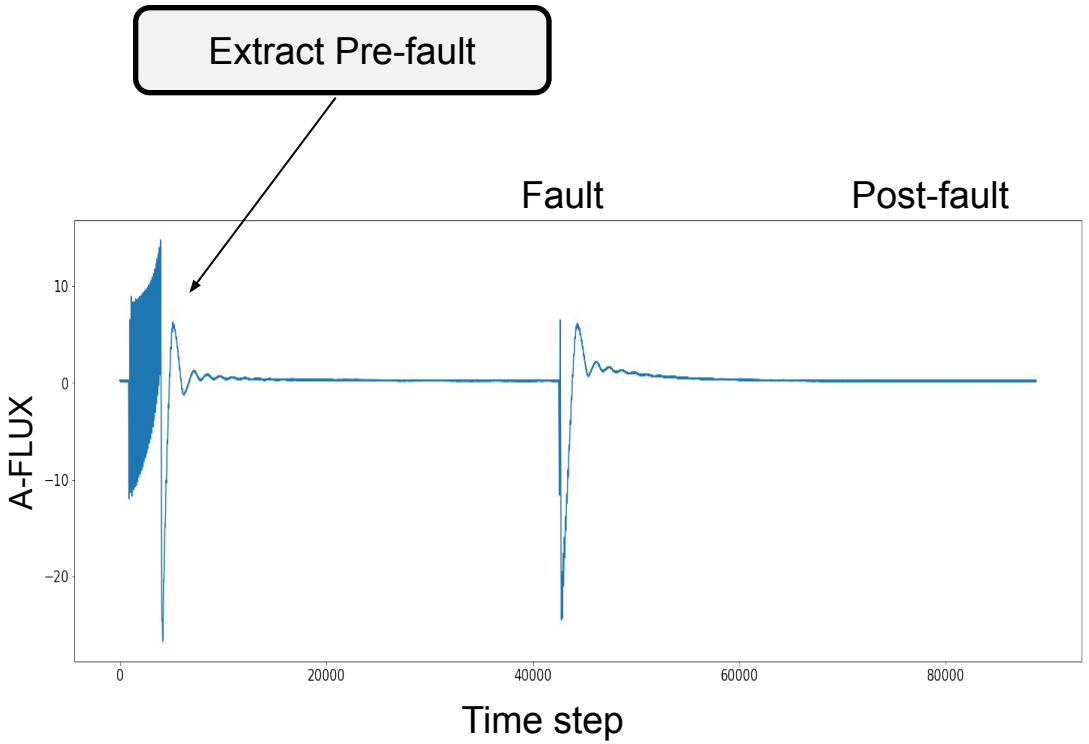

Normalized Density
0.15
0.10
0.05
IGBT
SCR
Driver
FLUX
A + IGBT − I
A − FLUX
MOD − V
Normalized Density
0.15
0.10
0.05
$10^{-7}$
$10^{-5}$
$10^{-3}$
$10^{-1}$
$10^{1}$
MOD − I
CB − I
DV/DT


Normal waveforms feature map (MEAN)
Samples
Feature map
0.12
0.10
0.08
0.06
0.04
0.02


Faulty waveforms feature map (MEAN)
Samples
Feature map
0.12
0.10
0.08
0.06
0.04
0.02

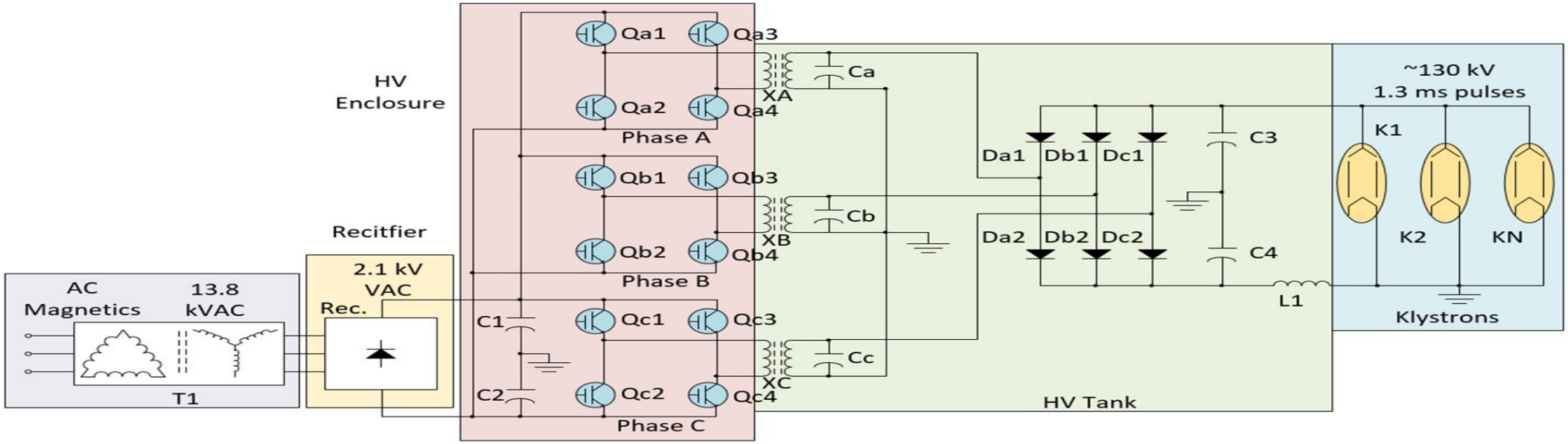
# Data Preparation

- Normal file: we extract all three micro-pulses and label it as "Normal".
  - Each 1.8 ms macro-pulse has 4500 time steps (i.e. sampling rate is 400 ns).
- Abnormal file: we extract the first micro-pulse (pre-fault) and label it as "Abnormal".
  - Only extract the pre-fault to allow detecting the anomalies ahead of time.
  - Each 1.8 ms macro-pulse has 4500 time steps (i.e. sampling rate is 400 ns).
- We do this for 14 different waveforms:
  - Three magnetic fluxes: A-FLUX, B-FLUX, C-FLUX
  - Six IGBT current waveforms: A+, A+*, B+, B+*, C+, and C+*
  - Two waveforms that represent the cab bank voltage
  - Two waveforms represent the modulator output voltage and current.



# Background: High Voltage Converter Modulators Operation

- AC is filtered and transformed to 2.1 kVAC, then converted to DC by a six pulse controlled rectifier.
- DC is filtered by capacitors C1, C2 used to store energy for the HV pulses. The DC is chopped by three IGBT based H-bridges Qa1-Qc4 switching at 20 kHz nominal.
- The 20 kHz pulses are stepped-up to high voltage and filtered by the transformers Xa-Xc and Ca-Cc.
- High voltage pulses are combined in parallel and full wave rectified by Da1-Da2. The resulting 120 Hz pulses are further filtered and feed to the cathode of the klystrons.






OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

Jefferson Lab

# Abnormal Data Descriptions

- We also extract the waveforms occur before the fault happens (abnormal data).
- There are several fault types across the modules which can be shown in Fig. 6.
- Similar to the normal data, we also extract 4500 timestep and 14 waveforms.
- We group fault types into 10 relative categories to increase the number of statistics.
- The grouped faults can be seen in Fig. 7, which represents all faults in all modules.

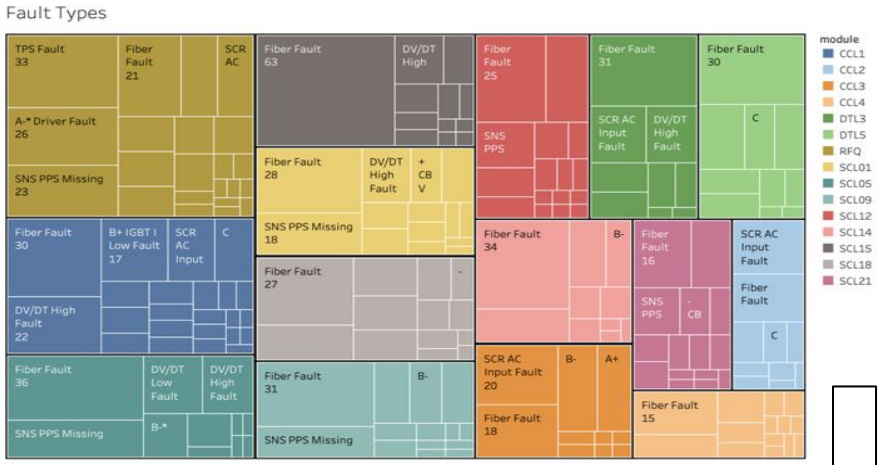



*Figure 6. Fault types for different modules*

Group faults into 10 categories

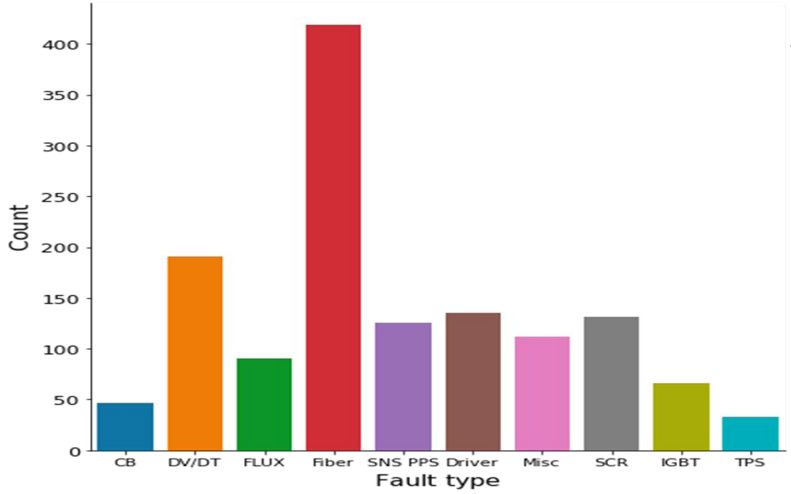



*Figure 7. Grouped faults*