

Application of convolutional and feedforward neural networks for fault detection in particle accelerator power systems

Majdi I. Radaideh*¹, Chris Pappas¹, Pradeep Ramuhalli², and Sarah Cousineau¹

¹ *Spallation Neutron Source, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, United States*
*radaidehmi@ornl.gov, pappasgc@ornl.gov, scousine@ornl.gov

² *Nuclear Energy and Fuel Cycle Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, United States*
ramuhallip@ornl.gov

ABSTRACT

High voltage converter modulators (HVCM) provide power to the accelerating cavities of the Spallation Neutron Source (SNS) facility. HVCMs experience catastrophic failures, which increase the downtime of the SNS and reduce beam time. The faults may occur due to different reasons including failures of the resonant capacitor, core saturation due to the magnetic flux, insulated-gate bipolar transistor (IGBT) failures, and others. We recently have setup a HVCM test stand to develop and test machine learning models for anomaly detection and fault prognostics. In this work, we propose binary classifiers and autoencoder architectures based on convolutional (CNN) and feedforward neural networks (FNN) to facilitate distinguishing normal from faulty waveforms coming from the HVCM during operation. The results indicate that the CNN binary classifier is the best model among the four showing very stable performance in the training and testing sets with impressive precision and recall metrics, reaching up to 99% with a very small uncertainty. The FNN classifier shows the least performance with a large uncertainty in its metrics. The performances of the two autoencoders based on CNN and FNN were in between, showing very good performance nonetheless.

Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>). Radaideh, M. I., et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL) accelerates protons to high energies which are used to produce a neutron beam used for neutron scattering (Henderson, 2014). The beam is accelerated in a linac consisting of a Radio Frequency Quadrupole (RFQ) section, a Drift Tube Linac (DTL) section, a Couple Cavity Linac (CCL) section and a Superconducting Linac (SCL) section. The accelerating cavities in each of these sections are fed by high power microwave amplifiers or klystrons. The klystrons are in turn powered by High Voltage Converter Modulators (HVCMs) which convert 3ϕ , 13.8 kVAC into a maximum of 135 kV, 1.3 ms long pulses at 60 Hz. The HVCMs can drive as many as 10 klystrons, depending on the klystron type and which section of the linac the klystron is located. There are a total of 15 HVCMs for SNS operations.

The HVCMs have been a significant source of lost user time at the SNS, as was shown by (Radaideh, Pappas, Walden, et al., 2022). The HVCM continues to experience system failures due to various reasons including IGBT switch failures, magnetic flux anomalies, and failures in the resonant capacitors. With these system failures, HVCM is ranked among the top sources of downtime for the SNS (Radaideh, Pappas, Walden, et al., 2022), which have led to several interruptions to the SNS user program. One of the solutions to improve the reliability of the HVCMs is to apply statistical or machine learning algorithms to detect impending failures or system anomalies in the waveforms collected from the HVCM controller. For example, previous study on using discrete cosine transform for anomaly detection in HVCM waveforms was done by (Pappas, Lu, Schram, & Vrabie, 2021), while (Radaideh, Pappas, Walden, et al., 2022) developed advanced recurrent neural network autoencoder models for time series anomaly detection in the HVCMs powering the RFQ section. Further efforts on applications of machine learning for fault detection in particle accelerators include application of vari-

ety of binary classifiers (Rescic, Seviour, & Blokland, 2020), Siamese neural networks (Blokland et al., 2021), adaptive neural networks for time-varying beam control (Scheinker, 2021), and similar others (Edelen et al., 2016). Overall, neural networks have demonstrated a promising potential in the field of fault identification and diagnosis as described in this comprehensive survey (Mohd Amiruddin, Zabiri, Taqvi, & Tufa, 2020).

The operating history with lost user time along with the large amount of data available make the HVCM a good candidate for machine learning. The major goal of the overall project is to be able to predict failures before they occur, and warn the SNS control room of impending failures and long term degradation of components such as metalized film capacitors which degrade over a period of years. In particular, this paper is a building block to achieve the primary project goal, where this paper focuses on the application of different neural network models to distinguish normal from faulty signals, which can be used to predict impending failures. The fundamental difference between this work and the previous studies (Pappas et al., 2021; Radaideh, Pappas, Walden, et al., 2022) is that we established an HVCM test stand instrumented to collect large amounts of waveform data to develop and test machine learning algorithms, which offers more data and more continuity in data streams than what was available before. Machine learning models are trained and tested to distinguish normal from faulty waveforms based on real data from the test stand. The models investigated in this work include two binary classifiers based on convolutional (CNN) and feedforward fully-connected neural networks (FNN). Based on CNN and FNN, two autoencoder (AE) models are also proposed and tested in this study (CNN-AE and FNN-AE).

2. EXPERIMENTAL SETUP

A simplified schematic of a HVCM is shown in Figure 1. Three-phase 13.8 kVAC line power is converted to ± 1300 VDC by the transformer T1 and a six pulse controlled rectifier circuit. This voltage is filtered with capacitor C1 and C2 which store sufficient charge to produce 1.3 ms pulses without excessive droop. The DC voltage is supplied to three IGBT based H-bridge circuits operating at a nominal switching frequency of 20 kHz (Reass et al., 2001). The three phases are switched with a 120° phase shift between the phases, and the high power pulses are stepped up to high voltage using pulse transformers. The leakage inductance of the pulse transformers form a resonant circuit with the secondary capacitors Ca to Cc in Figure 1, giving the circuit a gain which is frequency dependent. The high voltage bipolar pulses from the resonant capacitors are recombined and rectified by the diodes Da1 to Dc2, forming the output pulses with an apparent switching frequency of 120 kHz which is filtered by C3, C4 and L1 and applied to the cathode of the klystrons. The HVCMs are operated with the IGBT switching frequency be-

low resonance of the H-bridges, allowing for compensation of output pulse droop from the storage capacitor voltage and the magnetizing inductance of the pulse transformers. This is done by modulating the IGBT switching frequency from low to high during the pulse and can flatten the top of the pulse to better than $\pm 1\%$ full scale.

The HVCMs use PXI-based controllers to control the timing of IGBT gating and ensure some signal values such as IGBT peak, commutating currents, and volt-seconds of the pulse transformers stay in a safe range. The controller also is used to set warning and trip levels for a variety of signals, digitize and save waveforms and communicate via ethernet to the SNS control system. Tuning the HVCMs is done manually at present and involves setting start and stop frequencies for IGBT gating to minimize droop, varying the start timing of the initial gate signals to minimize the likelihood of saturating the magnetic transformers. There are precision timing controls to correct for ripple on the flat top, but this is not normally required for SNS operation. Tuning is performed by experienced technicians and involves making incremental changes at reduced power while monitoring multiple signals such as klystron voltage, IGBT currents and core flux to ensure they meet pulse requirements and remain within preestablished safe values. Tuning is normally done after maintenance on a particular HVCM and is not re-tuned until further maintenance.

In addition to the 15 modulators used for operating the SNS, three RF test stands are used for testing various high power RF systems such as the klystrons, accelerating cavities and different HVCM configurations. One of these test stands, the Radio Frequency Test Facility (RFTF), was chosen in this study to install an upgrade of the HVCM data acquisition system to be able to stream and save data for the machine learning effort. The Normal/Fault files archived for the SNS HVCMs require storage of approximately 30 MB of data when decimated to 2.5 MS/s. Because of hardware limitations with the present controller, the rate we can stream this data, and the massive amount of disk space required to store streamed data, the number of waveform channels was reduced from 32 to 12, and the record length was reduced from 3.6 ms to 1.5 ms. This reduced the size of each waveform file from 30 MB to approximately 540 kB.

Out of the 12 channels being saved, 10 of them are used in this work, which are the six IGBT current waveforms in the phases (A+, A+*, B+, B+*, C+, C+*), three magnetic fluxes in the three phases (A-flux, B-flux, C-flux), and the modulator current (Mod-I). Plot of 20 selected A-flux waveforms is shown in Figure 2 for normal cases and in Figure 3 for faulty cases. The waveforms are randomly selected from a large pool described in the next section. The fault data come from two main sources: (1) real fault events in the RFTF and (2) data collected during the HVCM tuning phase (fault-like

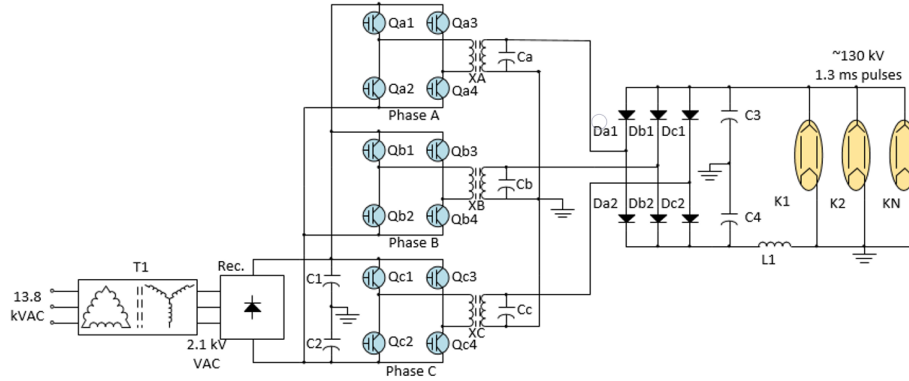


Figure 1. Simplified schematic of an HVCM

data). The second source dominates the faulty data given that real faults in the HVCM/RFTF do not occur very frequently and we recently started streaming data from the machine. As described before, HVCM tuning is done manually by the operators following a HVCM startup. This process involves tweaking the HVCM settings to different values to optimize the waveform shapes, and our data acquisition system is programmed to record pulses at the maximum saving rate (1 pulse per second) during the tuning process. These tuning waveforms deviate from normal operating ranges and can serve as a great source of abnormal conditions to increase the sample size of the fault data. In Figures 2-3, we can notice that the normal waveforms are quite similar, while the faulty ones could have shapes close to normal or absolutely erratic. In order to not magnetize the cores of the pulse transformers and avoid turning off at high current, the last IGBT current pulse is allowed to complete the conduction period, which means that an extra 1/2 cycle at the end of the pulse may occur at higher frequency to complete the cycle. This justifies why some pulses in Figure 2 have positive tails while others have negative ones.

Although some of those waveforms can be easily detected by eye by an expert, the rate and the time scale (1.5 ms) at which these pulses are saved make it extremely difficult to perform fault detection without an automated system like machine learning.

3. METHODOLOGY

The methodology is described in 3 subsections: the data-preprocessing procedure is described first, then the proposed machine learning models are described, followed by the performance metrics used to evaluate the models.

3.1. Data Preparation

We have collected a dataset containing about 20,000 normal waveforms and 5,000 faulty waveforms. Although it is easier to collect more normal data, collecting faulty data is much

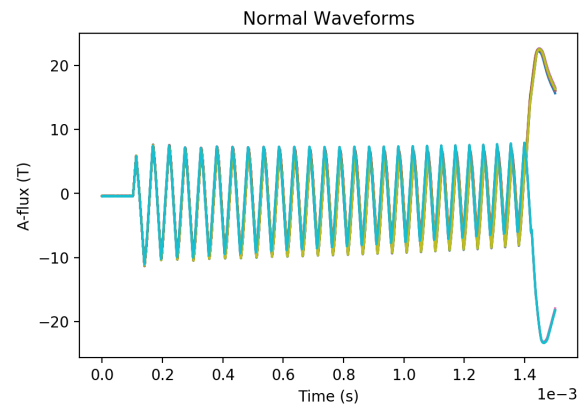


Figure 2. Normal magnetic flux waveforms in phase A

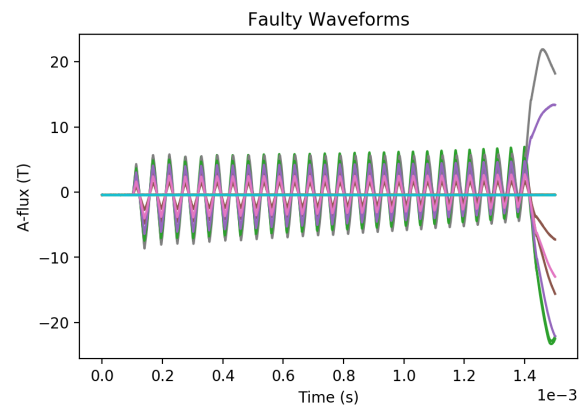


Figure 3. Faulty magnetic flux waveforms in phase A

more challenging until fault scenarios occur or when the operators perform HVCM tuning. The waveform data are parsed from a CSV format into numpy arrays.

To provide a solid testing of the proposed models, we withheld 2000 normal pulses and 2000 anomaly pulses as a test set. This leaves us with 3000 anomaly samples and about

18,000 normal samples for training. To remove the class imbalance between the normal and anomaly pulses, we randomly picked 3000 normal samples from the 18,000, and use them for training. As will be shown later in this study, the current data split ensures an excellent performance where larger datasets did not provide additional improvement.

The input data (X) represents the waveforms/pulses, which are a 3D tensor with the shape

$$shape^X = (N_{pulses} \times N_{times} \times N_{features}), \quad (1)$$

where N_{pulses} is the number of different pulses/samples collected from the system, $N_{times} = 3753$ is the number of time steps for each pulse, and $N_{features} = 10$ is the number of different features or waveform types recorded for each pulse, which were described in the previous section. The input data is scaled using a min-max scaler to locate between [0,1] to facilitate the training process.

For the classifiers (CNN, FNN), the label of the pulse (Y) is a binary vector of 1 or 0, referring to whether the pulse is normal or anomaly, respectively. For the AE (FNN-AE, CNN-AE), the labels are not explicitly provided to the model. Instead, the normal and anomaly pulses are separated in two independent datasets, where the AE is trained with the normal pulses to determine the reconstruction error threshold, and evaluated by the anomaly pulses to determine if the AE prediction exceeds that threshold.

3.2. Proposed Models

We propose and compare the performance of four different neural network models with four different architectures. The models are: (1) classical fully-connected feedforward neural network classifier (FNN), (2) convolutional neural network classifier (CNN), (3) feedforward neural network autoencoder (FNN-AE), and convolutional neural network autoencoder (CNN-AE). These models are described briefly, where grid search was used to determine the optimal configuration for each model.

3.2.1. FNN

The FNN classifier consists of fully-connected and dropout layers. The input shape to the network is a 2D flattened version of $shape^X$, where the time step and waveform axes are flattened to a single axis. The output is a binary prediction of the probability of a sample being normal or anomaly pulse. The architecture of the FNN classifier is as follows:

1. Dense layer with 64 nodes, ReLU activation, followed by 0.2 dropout.
2. Dense layer with 32 nodes, ReLU activation, followed by 0.2 dropout.
3. Dense layer with 16 nodes, ReLU activation, followed by 0.2 dropout.
4. Dense layer with 8 nodes, ReLU activation, followed by 0.2 dropout.
5. Dense layer with 2 nodes, Softmax activation.

3.2.2. CNN

The CNN classifier consists of Conv1D, max pooling, and fully-connected layers. The input shape to the network is a 3D tensor of $shape^X$. The output is a binary prediction of the probability of a sample being normal or anomaly pulse. The architecture of the CNN classifier is as follows:

1. Conv1D layer with 32 filters, 6x6 kernel, ReLU activation, followed by max pooling of size 2x2.
2. Conv1D layer with 32 filters, 4x4 kernel, ReLU activation, followed by max pooling of size 2x2.
3. Conv1D layer with 16 filters, 3x3 kernel, ReLU activation, followed by max pooling of size 2x2.
4. Conv1D layer with 8 filters, 2x2 kernel, ReLU activation, followed by max pooling of size 2x2.
5. Flatten layer.
6. Dense layer with 64 nodes and ReLU activation.
7. Dense layer with 16 nodes and ReLU activation.
8. Dense layer with 2 nodes and Softmax activation.

3.2.3. FNN-AE

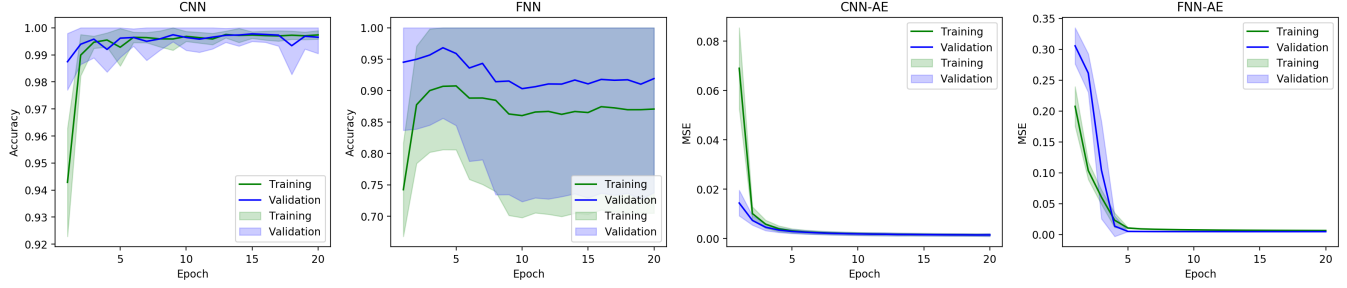
The FNN-AE consists of fully-connected and dropout layers, with layer/node size consistent with the FNN classifier. The input and output shape are identical for FNN-AE, which is a 2D flattened version of $shape^X$, where the time step and waveform axes are flattened to a single axis. The architecture of the FNN-AE is as follows:

1. Encoder: Four dense layers with 64, 32, 16, and 8 nodes, respectively. Each layer has ReLU activation and is followed by a 0.2 dropout layer.
2. Bottleneck: Dense layer with 4 nodes, ReLU activation.
3. Decoder: Four dense layers with 8, 16, 32, and 64 nodes, respectively. Each layer has ReLU activation and is followed by a 0.2 dropout layer.
4. Output layer: A dense layer with 37530 nodes (i.e. $N_{times} \times N_{features}$) and linear activation.

3.2.4. CNN-AE

The CNN-AE consists of Conv1D layers with kernel/filter size consistent with the CNN classifier. The input and output shape are identical for CNN-AE, which is a 3D tensor of $shape^X$. The architecture of CNN-AE is as follows:

1. Encoder: Three Conv1D layers with 32 filters 6x6 kernel, 32 filters 4x4 kernel, and 16 filters 3x3 kernel, respectively. Each layer has ReLU activation.

Figure 4. Training curves (mean $\pm 1\sigma$) for the four proposed models

2. Bottleneck: Conv1D layer with 8 filters, 2x2 kernel, ReLU activation.
3. Decoder: Three Conv1DTranspose layers with 16 filters 3x3 kernel, 32 filters 4x4 kernel, and 32 filters 6x6 kernel, respectively. Each layer has ReLU activation.
4. Output layer: A Conv1DTranspose layer with 10 filters, 6x6 kernel, ReLU activation.

3.3. Performance Metrics

We evaluate the performance of all fault detection models using four different metrics. All of them have a maximum value of 1, and they are desired to be larger. The first is precision

$$Precision = \frac{TP}{TP + FP}, \quad (2)$$

where TP is the number of true positive predictions, FP is the number of false positive predictions, TN is the number of true negative predictions, and FN is the number of false negative predictions. The second is recall, which indicates the true positive rate

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

The metric $F1$ provides a harmonic mean of both precision and recall

$$F1 = \frac{2TP}{2TP + FP + FN}. \quad (4)$$

Lastly, the ROC (receiver operating characteristic) curve shows the relationship between the true positive rate against the false positive rate at various threshold settings. The area under the curve (AUC) of the ROC curve provides an indication about model performance. AUC can be defined as (Fawcett, 2006)

$$AUC = \int_0^1 TPR(FPR) dFPR, \quad (5)$$

where TPR is the true positive rate and FPR is the false positive rate. $AUC = 1$ is a perfect model with a zero false positive rate, while $AUC = 0.5$ is a baseline random detector with 50/50 detection probability.

4. RESULTS

4.1. Test Settings

To test the model stability, we train each of the proposed four models 20 times, each time with a different random seed for the network parameters and with a different training and testing sets (i.e. size remains the same). The training and testing data are randomly sampled from the large pool. Therefore, the network performance is reported based on the statistics of the metrics achieved by the 20 models.

In each round, the four network models are trained with similar hyperparameters which include: validation split of 0.2, batch size of 32, 20 epochs, and Adam optimizer with 5×10^{-4} learning rate. The loss function for the autoencoders (CNN-AE, FNN-AE) is the mean squared error, while for the classifiers (CNN, FNN), the loss function is the sparse categorical crossentropy.

The hyperparameters of the four models (nodes, learning rate, batch size, etc.) were determined by running a parallel grid search, where networks with different architectures are trained and tested and the best configuration is selected.

We have used Tensorflow with GPU support using CUDA and CuDNN libraries for the implementation of all proposed models. All training and analyses were conducted on a GPU cluster with 8 NVIDIA A100 SXM4 40GB GPUs available at the Spallation Neutron Source of the Oak Ridge National Laboratory.

4.2. Training Results

The training/validation curves for the four models are shown in Figure 4, where each curve shows the mean as well as 1 standard deviation of the metric based on the 20 independent training rounds. The results clearly show a stable and very good performance for CNN, CNN-AE, and FNN-AE as their training metric (accuracy, MSE) converges to an acceptable value with a small error bar. Also, the training/validation subsets show consistent results, implying no overfitting. However, this is not the case for the FNN classifier, as it can be inferred from the large error bars in both the training and validation accuracy. This implies that the FNN classifier is less

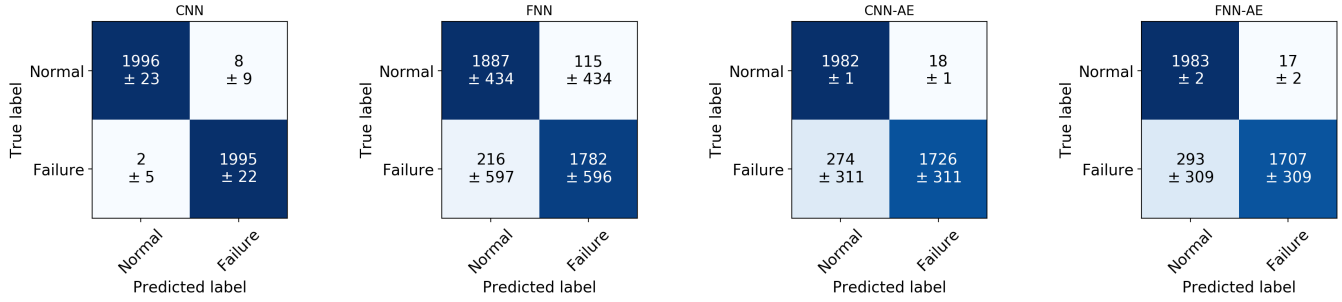


Figure 5. Confusion matrix (mean \pm 1σ) for the four proposed models (numbers are rounded to the nearest integer)

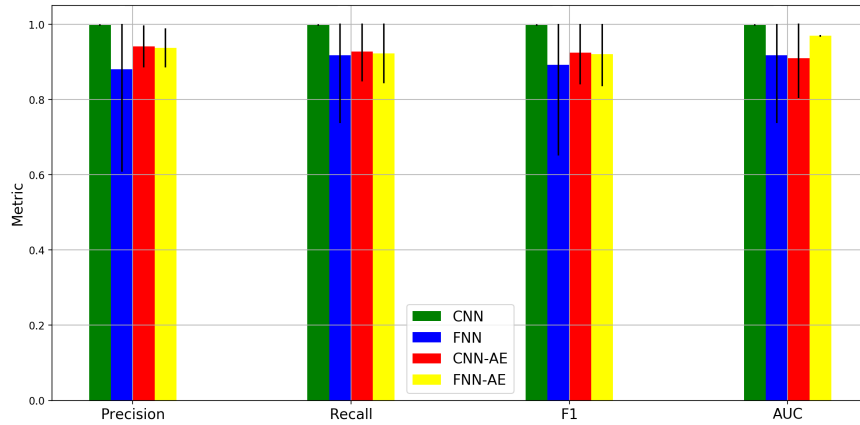


Figure 6. Performance metrics with uncertainty for the four proposed models

stable than the other three models, and could be dependent on the dataset being sampled as well as the initialization of the network parameters.

4.3. Testing Results

Applying the trained models on the test set to predict the pulse label gives the results in Figure 5 in the form of confusion matrix. The results are reported as the mean and 1 standard deviation of the class prediction (normal or failure). The confusion matrix shows that the CNN classifier is indeed the best among all four models predicting TP and TN with impressive accuracy, while maintaining FP (8 ± 9) and FN (2 ± 5) to very small numbers.

The next two best models are the two autoencoders (CNN-AE, FNN-AE), both with comparable performances. The TP and FP rates look very acceptable, which is not surprising, due to the autoencoder thresholding at 1% FP rate. However, the two autoencoders struggle at the FN, tagging close to 300 anomaly pulses as normal, also coming with a significant error margin (e.g. 293 ± 309).

The worst performing model in this study seems to be the FNN classifier, as we already observed from its unstable training in Figure 4. The confusion matrix in Figure 5 shows

high variability in predicting all four categories: TP, FP, TN, FN. This could imply that using the FNN classifier could lead to unreliable predictions.

4.4. Model Comparison

Based on the test set, the metrics described in section 3.3 are estimated and reported with uncertainty in Figure 6, while the numerical results are reported in Table 1. The metrics confirm the observations we noticed, the CNN is again showing the best performance for all metrics: precision, recall, F1, and AUC, with values close to 0.99 along with a very small uncertainty. Although FNN still shows fair metric values in average, their uncertainty is a bit large. Both autoencoders show comparable metrics, except that FNN-AE has slightly better AUC than CNN-AE, which is also with a smaller error bar.

In terms of computing time, when using the same GPU, the autoencoders (CNN-AE, FNN-AE) are slower to train than the binary classifiers (CNN, FNN) by a factor of 2, mainly because of the additional layers needed to have a symmetric encoder/decoder architecture. The CNN and FNN classifiers (and CNN-AE and FNN-AE) have comparable training times when compared to each other, with the CNN variant being slightly slower than the FNN variant (by 5-10 seconds).

Table 1. Model metrics with uncertainty (in parenthesis) as reproduced from Figure 6

Method	Precision	Recall	F1	AUC
CNN	0.998 (0.002)	0.998 (0.002)	0.998 (0.002)	0.998 (0.002)
FNN	0.88 (0.272)	0.917 (0.18)	0.892 (0.241)	0.917 (0.18)
CNN-AE	0.941 (0.056)	0.927 (0.08)	0.925 (0.085)	0.909 (0.105)
FNN-AE	0.937 (0.052)	0.922 (0.079)	0.92 (0.085)	0.969 (0.002)

We found that the effects of model hyperparameters on the performance are not significant, especially for hyperparameters like learning rate and batch size, which justify why we fixed them across all models. This can be explained mostly because of the quality and large amount of the data available to all models, which can reduce model sensitivity to hyperparameters. As a result, we should explain here that the poor performance of FNN has to do with the FNN classifier itself rather than its hyperparameters, since we observed that for a certain round for a given data split, the FNN classifier can perform well, while for another round and another data split, the same architecture can perform poorly.

4.5. Discussion

In this study, it is worth highlighting that the data were collected from the system and the model training and testing were done offline. Even though the data acquisition system was upgraded later after this work was conducted to facilitate near real-time data streaming, the topic of online training will be kept to when we extend this work to perform fault prognosis, i.e., predicting the fault ahead of time by tracking fault precursors (e.g., system degradation). Accordingly, this paper focused primarily on demonstrating the potential of machine learning for fault detection which could be helpful to improve HVCM reliability. The output signal from machine learning is restricted to a warning message sent to the operators once an abnormal waveform is identified. No automatic decision is made by machine learning at this stage since additional efforts are needed to install the trained models on the system either internally within the FPGA or externally after the data are streamed, while ensuring that these models do not interfere with the HVCM operation. In either case, the implementation part is currently under investigation by the team.

Given the previous model results, it is obvious that the subsampling we performed in section 3.1 to remove class imbalance between the normal and fault data had a small effect. This is explained by the fact that the CNN classifier has already achieved an excellent performance (see Figure 6), which implies that providing additional normal data will not improve the performance that much. In addition, collecting more normal data is much easier than collecting fault data, so adding more normal events to the dataset has never been an issue to the authors.

The results of this work provide a promising potential for the usage of machine learning to detect anomaly signals that can lead to catastrophic failures in the HVCM; resulting in a downtime for the SNS. This study is based on a test HVCM (RFTF) and not based on the main 15 HVCMs powering the SNS. The main observation found in this study compared to the previous study with RNN/LSTM (Radaideh, Pappas, Walden, et al., 2022) is that this study shows the effect of having large amount of streaming data on the performance of machine learning, where accuracy can reach up to 99% using less complex and less hyper-parameterized models. On the other hand, using limited waveform data such as the data used for the RFQ module, advanced and complex RNN models along with significant tuning were needed to achieve promising results by (Radaideh, Pappas, Walden, et al., 2022).

Therefore, it is worth highlighting an important difference between the data we used here (from the RFTF) and the data we published recently from the main 15 HVCMs powering the SNS (Radaideh, Pappas, & Cousineau, 2022). The previously published data (Radaideh, Pappas, & Cousineau, 2022) have many fault events recorded from multiple modules, however, the data are not continuous in time as only the pulse before the fault event is available, i.e., that data cannot be used for prognosis applications but can be used for multi-class fault classification. The current RFTF data are steamed with a much better time continuity, where system configuration and settings remain almost the same, which make them a good fit for prognosis even though the number of fault sources/varieties is very limited, i.e., cannot be used for multi-class classification. The team plans to share the RFTF dataset used in this work with the community very soon after our extended work on prognosis is approved.

5. CONCLUSIONS

We have established a test facility in the spallation neutron source (SNS) to explore machine learning models for fault detection in the high voltage converter modulators that power klystrons in particle accelerators. Four models are investigated including two binary classifiers based on convolutional (CNN) and feedforward (FNN) neural networks, and two autoencoder models (CNN-AE, FNN-AE) based on the same network types. The results indicate that the CNN binary classifier is the best model among the four showing very stable performance in the training and testing sets with impressive metrics of precision and recall reaching up to 99% with a

very small uncertainty. The FNN classifier shows the least performance with a large uncertainty in its metrics, illustrating sensitivity to the data being selected for training and the initialization of the network weights (i.e., overfitting). The performances of the two autoencoders were in between.

The extension of this work will include a field application of the proposed models, where the authors will test the pre-trained models in predicting the impending failure of the system earlier by detecting some anomalous waveforms in advance to the fault event, and notifying the operators (i.e., fault prognosis). This will highlight the value of machine learning in impending fault detection by utilizing the trained models over real-time data streams.

ACKNOWLEDGMENT

The authors are grateful for the support from the Neutron Sciences Directorate at ORNL in the investigation of this work. This work was supported by the DOE Office of Science under grant DE-SC0009915 (Office of Basic Energy Sciences, Scientific User Facilities program). This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

- Blokland, W., Ramuhalli, P., Peters, C., Yucesan, Y., Zhukov, A., Schram, M., ... Jeske, T. (2021). Uncertainty aware anomaly detection to predict errant beam pulses in the sns accelerator. *arXiv preprint arXiv:2110.12006*.
- Edelen, A. L., Biedron, S., Chase, B., Edstrom, D., Milton, S., & Stabile, P. (2016). Neural networks for modeling and control of particle accelerators. *IEEE Transactions on Nuclear Science*, 63(2), 878–897.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8), 861–874.
- Henderson, S. (2014). The Spallation Neutron Source accelerator system design. *Nucl. Instrum. Meth. A*, 763, 610–673. doi: 10.1016/j.nima.2014.03.067
- Mohd Amiruddin, A. A. A., Zabiri, H., Taqvi, S. A. A., & Tufa, L. D. (2020). Neural network applications in fault diagnosis and detection: an overview of implementations in engineering-related systems. *Neural Computing and Applications*, 32(2), 447–472.
- Pappas, G., Lu, D., Schram, M., & Vrabie, D. (2021, 08). Machine Learning for Improved Availability of the SNS Klystron High Voltage Converter Modulators. In *Proc. ipac'21* (pp. 4303–4306). JACoW Publishing, Geneva, Switzerland. doi: 10.18429/JACoW-IPAC2021-THPAB252
- Radaideh, M. I., Pappas, C., & Cousineau, S. (2022). Real electronic signal data from particle accelerator power systems for machine learning anomaly detection. *Data in Brief*, 43, 108473.
- Radaideh, M. I., Pappas, C., Walden, J., Lu, D., Vidyaratne, L., Britton, T., ... Cousineau, S. (2022). Time series anomaly detection in power electronics signals with recurrent and convlstm autoencoders. *Digital Signal Processing*, 130, 103704.
- Reass, W., Doss, J. D., Gribble, R., Lynch, M., Rees, D., Talerico, P., & Borovina, D. (2001). Operational results of the spallation neutron source (sns) polyphase converter-modulator for the 140 kv klystron rf system. In *Pacs2001. proceedings of the 2001 particle accelerator conference (cat. no. 01ch37268)* (Vol. 2, pp. 1029–1031).
- Rescic, M., Seviour, R., & Blokland, W. (2020). Predicting particle accelerator failures using binary classifiers. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 955, 163240.
- Scheinker, A. (2021). Adaptive machine learning for robust diagnostics and control of time-varying particle accelerator components and beams. *Information*, 12(4), 161.