SAND2020-5893C

DEVELOPING SIGNATURES-BASED SAFEGUARDS FOR ENRICHMENT FACILITIES

Nathan Shoman*
Sandia National Laboratories†
nshoman@sandia.gov

Philip Honnold
Sandia National Laboratories†
phonnol@sandia.gov

Benjamin B. Cipiti
Sandia National Laboratories†
bbcipit@sandia.gov

James Garner
Oak Ridge National Laboratory
garnerjr@ornl.gov

ABSTRACT

The International Atomic Energy Agency's (IAEA) current approach to safeguards at enrichment facilities relies on attended non-destructive assay (NDA) and weight measurements in addition to unattended methods such as the Online Enrichment Monitoring System (OLEM). These measurement systems are typically used to detect a specific safeguards problem. For example, OLEM and environmental sampling are used to detect changes in uranium enrichment. The use of machine learning could improve safeguards through better data utilization. A signatures-based framework that uses existing measurement systems in order to improve facility safeguards is developed. This approach can detect, locate, and provide information on severity of anomalies that occur in an enrichment facility. Unlike some other machine learning approaches, the output of this approach is in an intuitive visual format. A balance-of-plant enrichment model was also developed through the course of this work to provide evaluation data for the machine learning based approach.

1 Introduction

The proliferation-relevant nature of technology and challenges associated with safeguarding gaseous centrifuge enrichment plants (GCEPs) have made the application of safeguards a high priority for both the United States and the IAEA[1]. The existing safeguards for GCEPs implemented by the IAEA have been fairly effective at detecting the diversion of declared material and the production of higher-than-declared enrichment. However, measurements used to detect one type of anomaly (such as the use of nuclear material accountancy to detect a diversion of declared material) could be used to enhance the detection of a different anomaly (higher-than-declared enrichment) by using data fusion and machine learning.

This work implements a modified version of the Multi-Scale Recurrent Encoder Decoder (MSCRED)[2] to detect anomalies at an enrichment facility. This algorithm has been shown to produce state-of-the-art anomaly detection probabilities on a set of real-world power plant data. The algorithm is also intuitive which could avoid some challenges faced by other machine learning based approaches. Generally, the algorithm uses the concept of signature matrices, which contain information on pair-wise relationships between facility signals, combined with machine learning to detect subtle anomalies within the facility. In addition to anomaly detection the algorithm can also provide some information on the anomaly's location and severity. This approach could be beneficial to the facility operator as well who could monitor for process upsets.

*Corresponding author

†Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

A generic balance-of-plant enrichment model was used to generate data in order to evaluate and train the algorithm. The following sections detail this model as well as the machine learning approach. This concept is still in development so discussion is restricted to general trends and observations rather than a strict quantitative comparison between this approach and traditional safeguards.

2 GCEP Facility Background

Many uranium-based nuclear applications require greater weight-percent of ^{235}U than is available in natural uranium. While multiple methods of enrichment exist, the most established and economical is enrichment by gas centrifuge.

An enrichment facility, regardless of type, can be considered, generally, as a "black box," where feed is used to create a higher enriched product stream and lower enriched tails stream. The capacity of an enrichment facility is described by Separative Work Units (SWUs) per year, where a SWU relates to the amount of work available to separate feed (*F*, with assay x_f) into product (*P*, with assay x_P) and tails (*T*, with assay x_T) streams.This concept is useful in relating the energy consumption of a facility with it's enrichment capabilities. The amount of material processed (given by F, P, T) is a function of the associated assays (x_F, x_P, x_T) and the available capacity (W_{SWU}).

$$W_{\text{SWU}} = P * V(x_P) + T * V(x_T) - F * V(x_F) \tag{1}$$

Where $V(x)$ is the value function, given by:

$$V(x) = (1 - 2x) \ln \frac{1 - x}{x} \tag{2}$$

In practice, a GCEP is comprised of hundreds or thousands of centrifuges. The above principles can be applied to a specific centrifuge. W_{SWU} is dependent on centrifuge parameters (size, rotation rate, etc.), and feed characteristics (*F*, x_F) which are typically known. Enrichment and mass balance relations can then be utilized to analytically determine enriched and depleted stream characteristics. To achieve a target enrichment, the centrifuges are arranged in a specific sequence. This is referred to as the centrifuge cascade. Facility-specific characteristics are typically sensitive information and not available for public disclosure. However, the principles described above can be applicable for an entire cascade as well as for individual centrifuges.

3 Development of a balance-of-plant enrichment model

Large-scale facilities are typically comprised of multiple cascades operating in parallel, within a cascade hall. A single UF_6 handeling area will service an entire hall. Each handeling area contains several different unit operations, including feed stations (hot boxes), withdrawal stations (cold boxes), autoclaves (for homogenizing and sampling), and accountancy scales (high-precision load cells).

Feed stations convert cylinders of solid UF_6 to a gas via sublimation, which is then split between multiple cascades. Since the cascades require tight tolerances for efficient operation, pumps and heaters are located before the cascades to adjust the thermophysical properties of the feed gas. Each cascade has an enriched uranium stream and a depleted uranium stream. These material flows are re-combined before heading to the withdrawl station where product and waste cylinders are filled. Currently waste cylinders that are filled with depleted UF_6 are stored on-site indefinitely. Product cylinders that are filled with the enriched UF_6 are weighed, sampled, and re-weighed before being packaged and shipped to customers.

3.1 BOP enrichment model details

Simulation of an enrichment facility poses a unique challenge, as it requires modeling both discrete and continuous signals. In-process UF_6 flow is best represented by continuous modeling, while the events associated with cylinder processing are best considered as discrete events. Matlab Simulink [3] and SimEvents were used to model the facility, drawing upon past experience with SSPM [4, 5].

Matlab Simulink is a model-based design code, commonly utilized for signal processing and simulation, but incorporating the full suite of Matlab libraries for addressing unique modeling constraints. Matlab SimEvents is a specific toolbox within Simulink specifically designed for event-driven discrete modeling. An overview of the facility flowsheet is shown in Figure 1

Before running the model, an initialization script is utilized in the Matlab Workspace. Facility size (in t-SWU/yr), product enrichment (in $\%$ ^{235}U), and tails (depleted) enrichment (in $\%$ ^{235}U) are input by the user. Feed material is

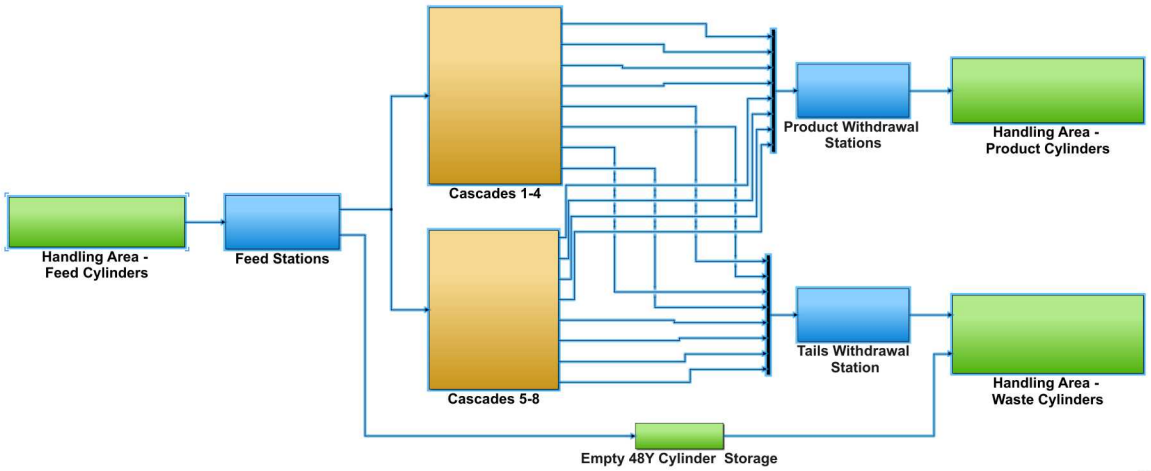

Figure 1: Enrichment model overview

assumed to be natural UF_6 (0.711% weight percent ^{235}U). The model has 4 feed stations, 4 product withdrawal stations, 4 waste withdrawal stations, and 8 parallel cascades. The operational year for this facility is assumed to be 8760 hours.

After running the initialization script, the model begins with modeling two different types of cylinders as discrete signals. The first cylinder type is a 48Y cylinder which can hold up to 12,500 kg of material and usually holds feed and waste UF_6. These cylinders can often times be reused as a GCEP will have hundreds of these cylinders on site at any given time. The second type is 30B cylinders which can hold about 2,250 kg of material and is used to hold the enriched UF_6 product.

These cylinders are simulated using SimEvents, which attaches a set of attributes to discrete entities. For 48Y cylinders, attributes include a random numeric designation, the weight of material to be removed, tare weight (weight of the empty cylinder), heels weight (amount of material left in cylinder after feed withdrawl), and enrichment of contents. For 30B cylinders, the attributes are: a numeric designation, weight of material to be received, tare weight, and enrichment of contents.

The feed cylinders entering the facility are generated at a regular interval specified by the user. These feed cylinders are assumed to be filled to $90\% \pm 4\%$ of actual capacity. The cylinders are weighed using an accountancy scale (high-precision load cell), emptied into the cascade hall at a feed station, and then re-weighed at an accountancy scale.

High precision accountancy scales are used to measure both feed and product cylinders. The scales measure the total weight of the cylinder and subtract the previously measured tare weight. A simulated error is added to the weight to simulate measurement error. The total error is set as the sum of systematic error and random error. Currently, the systematic and random errors are each modeled as 0.05%.

The individual feed and withdrawl stations are assumed to operate at a constant rate so that the facility is not bottlenecked. These stations simulate the transition from SimEvents-based discrete modeling to Simulink-based continuous modeling. The isotopic mass (F, ^{238}U, ^{235}U, ^{234}U) and thermophysical properties (pressure and temperature) are tracked for the gaseous UF_6 streams.

Gaseous UF_6 from the feed station travels to a vacuum pump before entering one of the 8 parallel cascades that are modeled. This pump simulates a pressure and temperature jump with random process variance to model the near-vacuum conditions of the cascade. The mass flow is split equally between the individual cascades with uniform thermophysical conditions. The individual cascades are treated as a "black-box". The product and tails enrichments are taken from the user-input in the initialization script, and outgoing mass flow rates calculated using the generalized equations and principles outlined in Section 2.

Material leaving the cascade proceeds through a post-cascade header pump, which is placed on both product and waste streams. These pumps simulate a change in thermophysical properties that correspond to moving from near vacuum conditions to more ambient conditions. The continuous operation of a near-vacuum condition is expensive and there is no need to maintain such conditions after the material has left the cascade. These thermophysical changes are again estimated from phase diagrams.

The product and waste streams are recombined before arriving at withdrawl stations. These stations convert continuous mass flows back into discrete entities. Material moving into these systems is summed, and upon reaching a given threshold ($90\% \pm 4\%$ net weight for 48Y or 30B), a cylinder is considered "full," and moved from the withdrawal station. As with feed stations, the material flow is assumed to be constant, such as to not be a simulation bottleneck.

As mentioned previously, 48Y cylinders are used for the waste material. Some of these may be new cylinders with no heel material, whereas others are re-used from empty feed cylinders. No further accountancy (other than a simple count) is performed on waste cylinders. Unlike the waste material, the product is loaded into 30B cylinders. After being filled, these are taken to accountancy scales, sampled at an autoclave, and then re-weighed. As with the feed cylinders, accountancy scales are assumed to be high-precision load cells, where measurement uncertainty is on the full container weight.

Once a product cylinder is filled it is sampled at an autoclave. This operation slowly heats and liquifies the contents of a cylinder in order to homogenize the UF_6, from which a small sample(~5 kg) is taken. Once the sample has been removed the cylinder contents are allowed to re-solidify and is re-weighed on the accountancy scale. The cylinders are then stored or shipped.

The model records time-series data in either excel or comma separated value format. Each dataset has a timestep of one hour and is generally a full year of operation in length (8760 rows). In-process flows consist of 6 features (4 mass flow features and 2 thermophysical features) which are recorded at a variety of locations. Cylinder features are also recorded on an hourly basis. Cylinder features are discrete and will only have a non-zero value at some observation times. The cylinder features include the designation, actual weight (ground truth), and measured weight. Samples taken from the product cylinders have features that include the corresponding cylinder designation, mass, and enrichment with incorporated measurement uncertainty.

4 MSCRED Algorithm development and modification

The detection of anomalies in multivariate time series data remains challenging and a subject of ongoing research. One of the principal difficulties is capturing the temporal nature of anomalies and determining the relevancy of historical data. Work by Song[6] has shown that the concept of signature matrices can be used to solve some of these issues. Specifically, these signature matrices encode both temporal dynamics in addition to correlations between time series. The concept behind a signature matrix applied to an enrichment facility is shown in Figure 2.

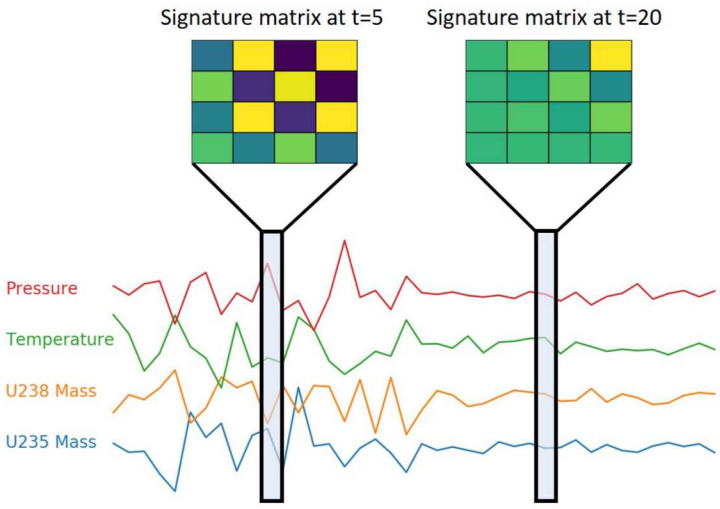

Figure 2: Example of the signature matrix concept

Signature matrices are constructed to form temporal correlations between multivariate signals. The example in Figure 2 has four signals that are monitored; ^{235}U mass, ^{238}U mass, pressure, and temperature. The relationships between these signals can be represented as a 4x4 signature matrix. Current work utilizes the signature matrix formulation proposed in the original MSCRED paper [2]. Pair-wise inner products of two time series from time $t-w$ to t can be represented by constructing an $n \times n$ signature matrix. Given two time series $x_i^w = (x_i^{t-w}, x_i^{t-w-1}, ..., x_i^t)$ and

$x_j^w = (x_j^{t-w}, x_j^{t-w-1}, ..., x_j^t)$ in a multivariate time series segment X^w their correlation m_{ij}^t is calculated as Equation 3.

$$m_{ij}^t = \frac{\sum_{\delta=0}^{w} x_i^{t-\delta} x_j^{t-\delta}}{\kappa} \tag{3}$$

where κ is a rescale factor, typically (κ=w). Currently, several different signature matrices are constructed at each timestep to capture anomalies with different duration. Signature matrices with shorter windows (eg. w in Equation 3) are designed to detect shorter, abrupt anomalies whereas matrices with longer windows can better detect longer, more protracted anomalies. Currently, signature matrices are constructed over 100, 400, and 800 hour windows. In the future, this could be adjusted by feedback from ongoing work.

As described in Equation 3, the feature matrix is usually a $n \times n$ matrix. The enrichment model outputs approximately 320 features, however, not all of these will be used in the current work. This analysis will focus on anomaly detection with only estimated mass and thermophysical properties, which consists of about 300 features. However, about 140 features are removed for being linear dependents of other features. The final signature matrix used in the current work is of shape 160×160 which contains mass and thermophysical properties from simulated measurements throughout the model.

Once these signature matrices are created they are then passed to the MSCRED algorithm. The algorithm attempts to learn the time-dependent behavior facility before identifying and pinpointing anomalies in the dataset. The MSCRED algorithm is a unique take on the well known autoencoder. The autoencoder architecture tries to learn a function $h_{W,b}(x) \approx x$. This forces the algorithm to learn the identity function with some constraint (limits on the number of terms in the function). In an autoencoder layers compress the data forcing each layer to learn a compressed representation of the data. The decoder then attempts to reconstruct the data given some compressed representation. Autoencoders can perform well for unsupervised anomaly detection as the weights are adjusted in training to reconstruct normal data. If the data is anomalous then the network should produce a poor reconstruction.

The MSCRED consists of a traditional encoder and decoder structure comprised of convolutional [7] layers. However, the architecture also includes attention layers that can help learn temporal trends. This attention layer serves to determine the importance of different temporal features, as not all features have equal value at predicting the current step. While the encoder and decoder layers can be implemented with existing Python libraries, the attention layer was specifically crafted for this algorithm and differs slightly from traditional Luong-style[8] and Bahdanau-style [9] attention.

4.1 Attention Layer

Sequential inputs to the autoencoder are temporally dependent on the previous timestep signature matrices. That is, the signature matrix calculated at $t = 100$ is temporally dependent on the signature matrix from $t = 99$. Consequently, the feature maps generated by the convolutional encoder is temporally dependent on previous time steps. The Convolutional Long-Short Term Memory (ConvLSTM) [10] was created to address deteriorating temporal performance in images over long sequences. The authors of MSCRED use the ConvLSTM to select feature maps from past states that are relevant to the current timestep. A temporal attention mechanism was developed to correlate the current state $\mathcal{H}^{t,l}$ relevant to past steps and aggregate the representations of those states to form a more refined feature map $\hat{\mathcal{H}}^{t,l}$. The refined feature map is formally defined as Equation 4.

$$\begin{aligned} \hat{\mathcal{H}}^{t,l} &= \sum_{i \in (t-h,t)} \alpha^i \mathcal{H}^{i,l} \\ \alpha^i &= \frac{\exp(\frac{\text{Vec}(\mathcal{H}^{t,l})^T \text{Vec}(\mathcal{H}^{i,l})}{\chi})}{\sum_{i \in (t-h,t)} \exp \frac{\text{Vec}(\mathcal{H}^{t,l})^T \text{Vec}(\mathcal{H}^{i,l})}{\chi})} \end{aligned} \tag{4}$$

5 Setup and Initial Results

The initial results consider the qualitative performance of the current MSCRED implementation. The following sections consider two different material loss scenarios of different magnitude and duration. The easiest case consists of a scenario where material is diverted over a shorter length of time. Conversely, the most difficult case is when material is diverted over longer periods of time.

This initial evaluation used signature matrices over several different temporal windows, as described in Section 4. Signature matrices that are summed over longer periods of time should have an easier time detecting more protracted material losses.

The general measure of abnormality for this algorithm is the number of pair-wise correlations that are poorly reconstructed, that is, the number of outputs from the algorithm with high reconstruction error. The original MSCRED algorithm set a threshold for the pair-wise errors and considered observations over that threshold to be anomalous. The evaluation of this approach and determination of the threshold is ongoing work. A probability of detection cannot yet be determined since the alarm threshold has not been finalized.

The following subsections describe initial results that detail the algorithm's performance as it relates to anomaly detection in both space and time. Many of the results and observations will be related to the algorithms performance on producing the correct signature matrix. Figure 3 shows an example of a signature matrix reconstruction during normal operation. For example, the coordinates (30,40) in Figure 3 would then refer to the pair-wise correlation, defined by Equation 4, that is reconstructed by the algorithm. The color of the specific square within the grid refers to the mean squared reconstruction error. Note that the reconstructed signature matrix error for normal operation is quite low, as indicated by the color bar.

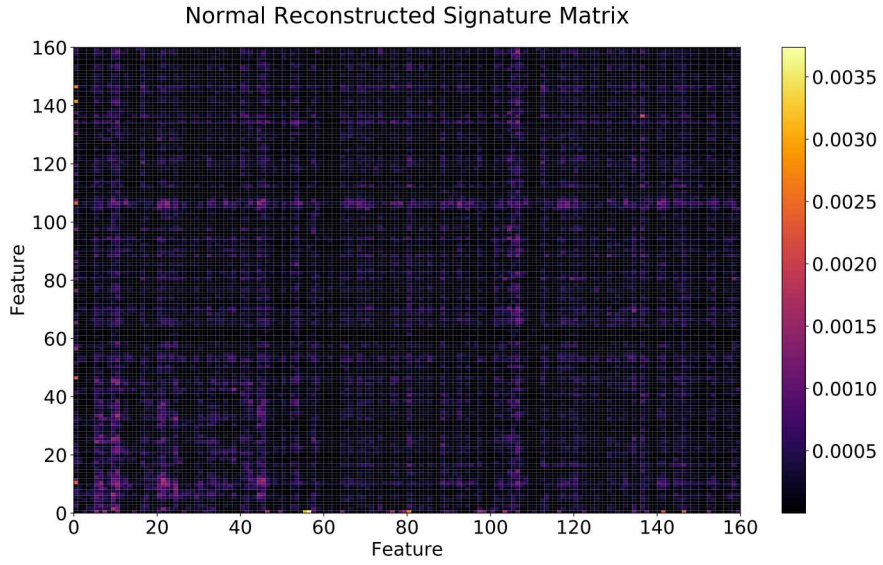

Figure 3: Example of a reconstructed signature matrix during normal operation

Each signature matrix represents a single slice of time which is generated at the same frequency at which the unattended measurements are made. This work assumes hourly measurements which means about 730 signature matrices would be calculated each month. These matricies have low computational overhead and can be calculated in a matter of seconds.

5.1 Abrupt Case

This first case considers the response of the algorithm to a simple abrupt material loss from the facility. The reconstructed signature matrix is shown in Figure 4. High reconstruction error bands are present in several features of the signature matrix , shown as bright colors in Figure 4, with much higher error than what is typically seen in the normal case, such as in Figure 3. The bands of high reconstruction error correspond to the location where the anomaly is occurring. This response provides an intuitive method for determining where an anomaly is occurring, and provides some information about the magnitude. Higher reconstruction errors correspond to a more serious deviation from the expected normal behavior. In practice, this would provide information for targeted follow-on inspections at enrichment facilities.

5.2 Protracted Case

The second case considers a more protracted material loss. The reconstructed signature matrix is shown in Figure 5. Again, note the higher reconstruction error bands which are presented as a bright color. These bands indicate that there is an anomaly at the location that corresponds to feature 60, 61, and 63. The signature matrix shows a strong sensitivity

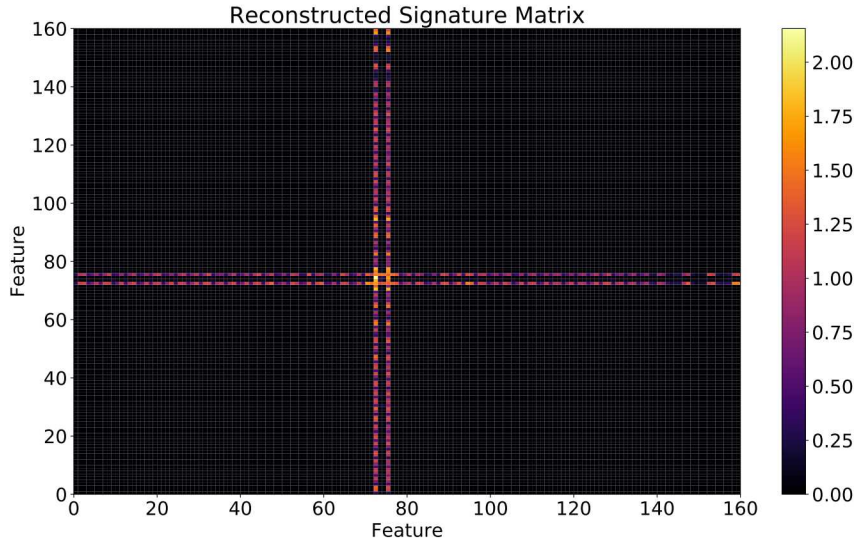

Figure 4: Signature matrix error during abrupt loss scenario

to the material loss, however, note the magnitude of the reconstruction error, described by the color bar in Figure 5. The protracted case has a lower reconstruction error than the abrupt case. This could provide inspectors or operations information regarding the severity of the anomaly.

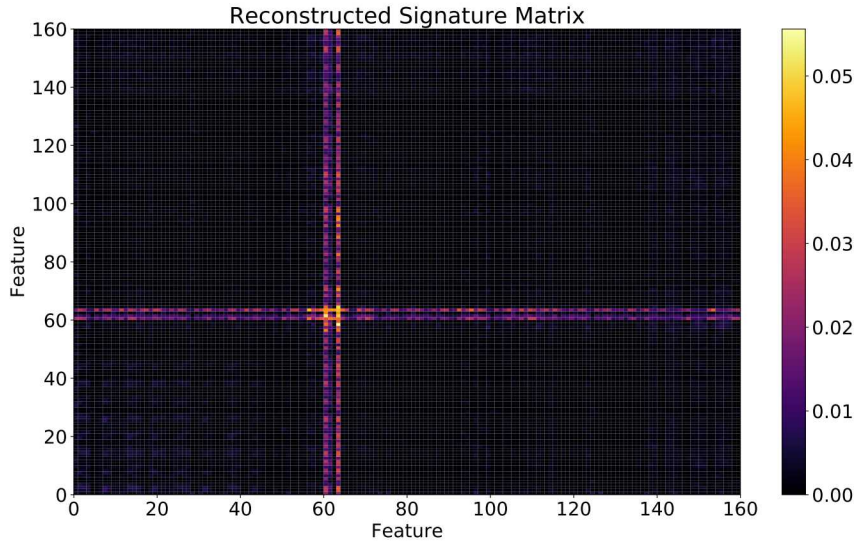

Figure 5: Signature matrix error during protracted loss scenario

6 Remarks on improving algorithm performance

While the results in the proceeding sections show good responses to the different anomalies, other examples do not. The ultimate goal is to mature the algorithm and produce a probability of detection. Ideally the algorithm will exhibit a high probability of detection for many different anomalous scenarios, however, it is impossible to detect all anomalies with perfect accuracy. Figure 6 shows an example where the algorithm exhibits next to no response to the same anomaly as

was present in Figure 4. It is expected that the algorithm will miss some anomalies, it is unusual that the algorithm barely responds to the anomaly at all.

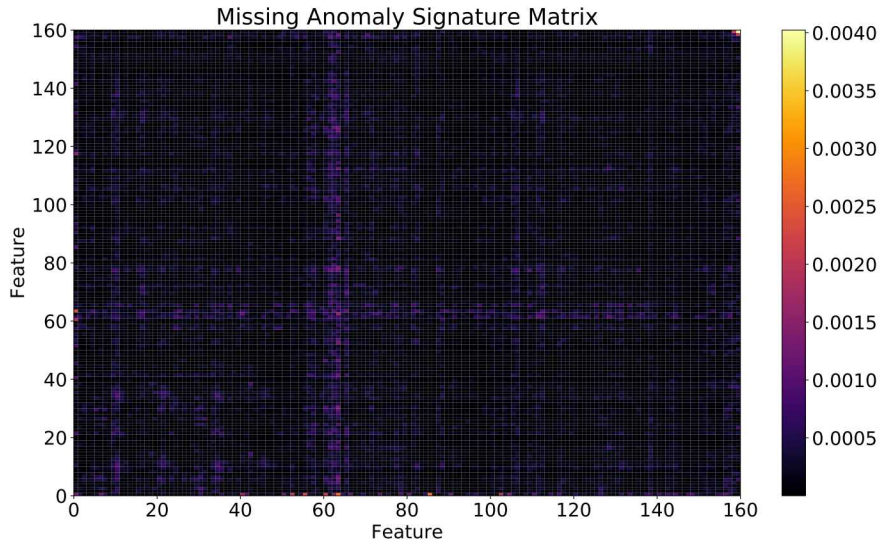

Figure 6: Signature matrix is reconstructed well even during a significant anomaly

For each of the two scenarios presented in this work the underlying balance-of-plant enrichment model is run 100 times with a different random seed. Each run will generate a slightly different signature matrix under the same conditions due to the variation of the in plant operations and randomness in the system. The initial implication of Figure 6 is that since the anomaly was reconstructed with low error, that particular signature matrix appeared as normal to the algorithm.

Concise and intuitive interpretation of the signature matrices over time is difficult which makes troubleshooting the algorithm challenging. At each time-step there are three different signature matrices, each of which is a 160×160 shaped matrix. Common techniques including Principal Component Analysis (PCA) [11] and t-distributed stochastic neighbor embedding (t-SNE) [12] were used to visualize the signature matrices over time to help diagnose and improve MSCRED performance.

PCA is similar to the encoder portion of the autoencoder in that it attempts to learn a lower-dimensional representation of higher-dimensional data. The primary differences between the encoder in MSCRED and PCA is the resulting dimensionality and the mapping used to transform the data. First, PCA uses a linear mapping to convert the 160×160 signature matrix to a lower dimension whereas the encoder uses a non-linear function. Secondly, the result of the encoder in MSCRED is still retains a relatively high dimensionality whereas PCA has a much lower dimensionality.

For this work, PCA is used to reduce the 160×160 signature matrix to a more intuitive two-dimensional space. PCA was used to confirm that the anomalies that were predicted as normal by the MSCRED algorithm, such as in Figure 6, could be viewed in a lower dimensional space.

PCA was then used to convert the lower-dimensional mapping back to the original 160×160 signature matrix. It was discovered that the error between the reconstructed PCA matrix and the original was quite high for the anomalous regions. In fact, a particular run that showed little error in MSCRED, shown in Figure 6 had high reconstruction error for the PCA reconstruction, shown in Figure 7. For certain cases the use of PCA as an autoencoder outperformed the MSCRED algorithm. It is unlikely that PCA outperformed the MSCRED strictly because it used a linear mapping. This observation lends support to insufficient compression in the MSCRED algorithm for good anomaly detection. That is, it is likely that PCA compresses the signature matrix more than MSCRED, which is key to detecting anomalies more effectively. In the future, the MSCRED algorithm will be modified to have more compression in the encoder layer.

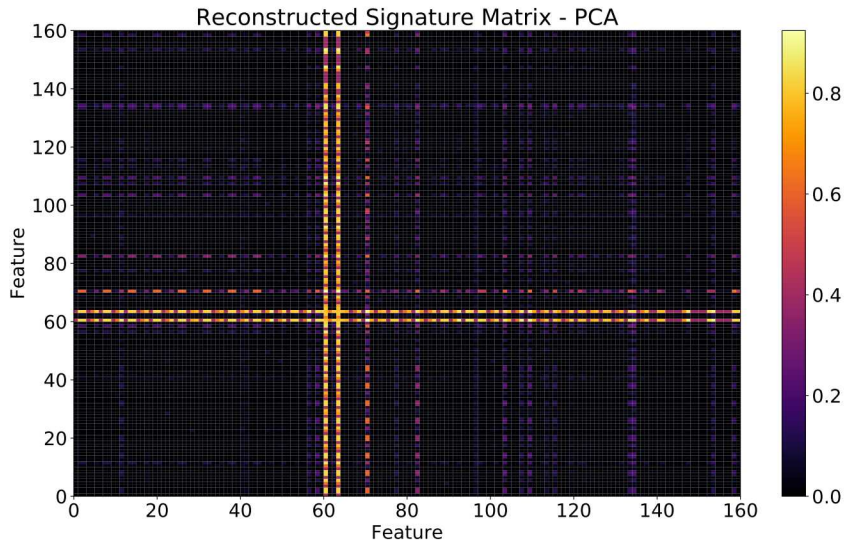

Figure 7: Signature matrix reconstructed with PCA

7 Conclusions and future work

This work demonstrates the proof-of-concept results for the utilization of a MSCRED-like algorithm with enrichment facility sensors. The results show that the algorithm can not only detect anomalies, but also locate them within a facility and provide some information about the magnitude of the anomaly. In the near future an alarm threshold, likely pair-wise reconstruction error, will be used to determine a probability of detection when using this algorithm. The machine learning approach can then be compared to traditional safeguards to evaluate whether this new approach can enhance safeguards using existing measurements. Future work will determine a probability of detection and consider other facility anomalies.

8 Acknowledgements

This work was funded through the National Nuclear Security Administration's Office of International Nuclear Safeguards. The authors would like to acknowledge the scikit-learn[13] machine learning API and Tensorflow[14] machine learning API.

References

[1] International Atomic Energy Agency. *Guidance for States and Implementing Comprehensive Safeguards Agreements and Additional Protocols*, 21 edition, May 2016.

[2] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, pages 1409–1416. AAAI Press, 2019.

[3] MathWorks. Simulink user's guide.

[4] Benjamin B. Cipiti and Nathan Shoman. Bulk handling facility modeling and simulation for safeguards analysis. *Science and Technology of Nuclear Installations*, 2018, 10 2018.

[5] Benjamin B. Cipiti. Process monitoring considerations for reprocessing. *Proceedings of the Institute of Nuclear Materials Management*, 2015.

[6] Dongjin Song, Ning Xia, Wei Cheng, Haifeng Chen, and Dacheng Tao. Deep r -th root of rank supervised joint binary embedding for multivariate time series retrieval. In Yike Guo and Faisal Farooq, editors, *KDD*, pages 2229–2238. ACM, 2018.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[8] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. pages 1412–1421. The Association for Computational Linguistics, 2015.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[10] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

[11] I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer New York, New York, NY, 1986.

[12] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.