An ML/AI Approach to Identifying Gaps in *a priori* Understanding of Nuclear Facility Design and Operations

Dan A. Rosa De Jesús, Lee Burke, Carlos González Rivera, Jackson Chin, Jereme Haack, Romarie Morales Rosado

Pacific Northwest National Laboratory, Richland, WA, USA

Abstract

Monitoring and characterization of nuclear facilities is an essential activity of nuclear non-proliferation, materials control, and safeguards. Such inferences are best supported by extensive *a priori* knowledge of facility design and operations, but that knowledge is not always correct, current, and complete. We present a technique for identifying discrepancies between *a priori* understanding and actual conditions on the ground by comparing the output of computational models of facility activities with sensor data gathered on-site. The technique leverages a novel unsupervised machine learning algorithm to provide a near-real-time rating of the discrepancy between expected and observed behavior. The algorithm is validated against a comprehensive anomaly detection benchmark, including 14 other unsupervised anomaly detection methods on ten datasets. We present promising results from applying the proposed technique to a prototype ML/AI system deployed at two testbed facilities. The results show the algorithm's effectiveness in identifying and explaining real-world discrepancies in support of monitoring and characterization activities.

1. Introduction

Monitoring and characterizing nuclear facilities are important activities for nuclear non-proliferation, materials control, and safeguards [1]. These serve as confidence-building measures and facilitate other responses by the international community when and if necessary. Over the beginning of this century, these activities have strengthened in key areas thanks to technological advances such as remote sensing and Machine Learning (ML) [2]. ML can be used to identify previously unknown entities of elevated risk through sensor fusion and, in the process, save hundreds of analyst hours in contrast to previous manual efforts. Still, inferences are best supported by extensive a priori knowledge of facility design and operations involving subject matter experts, but that knowledge is not always correct, current, and complete [3]. The discrepancies between a priori understanding and actual conditions on the ground can build bias into ML models and produce prejudiced inferences due to erroneous assumptions. A possible solution is to compare current foundational knowledge with the actual ground conditions to identify points in time that do not conform to a well-defined "normal" behavior.

Anomaly detection is an active area of research in cyber intrusion, fraud, industrial damage, image processing, and sensor networks [4]. In remote sensing, several challenges exist concerning the high dimensionality and diversity of the collected pieces of information, as well as the presence of missing values and other types of corruption inherited from the intermittent collection and communication of information. Many data processing techniques in conjunction with anomaly detection methods have been proposed to alleviate these issues. For instance, a semi-supervised deep hypersphere method combined with a deep neural network is proposed to separate anomaly features from regular features and identify anomalies [5].

Several strategies, including imputation, reduction, and marginalization, have been investigated to handle missing values in the context of anomaly detection [6]. Experiments on three anomaly detection methods and several datasets of different dimensionalities, percentages of anomalies, and modalities shed light on the conditions models and datasets must comply with before applying the strategies. However, the study exposes the limitations of applying these strategies. For instance, relevant information may be lost, and unwanted bias may be inserted into the data. Per the ethical and regulatory requirements of more robust solutions to these challenges, it is imperative to understand the individual contribution of different features towards anomaly identification to provide explanations to stakeholders of ML systems used in critical domains.

In the past, explainable anomaly detection methods have leveraged techniques, including feature projection, time-series decomposition, and Gaussian processes [7], to produce local explanations at the sample level. Yet, their transparency comes at a high computational cost and at the expense of only processing low-dimensional data. This paper proposes an unsupervised anomaly detection algorithm to alleviate the abovementioned challenges. The proposed approach combines the Self-Organizing Maps (SOM) with the k-Nearest Neighbors (kNN) for anomaly detection. The SOM model and kNN algorithm implement the Heterogeneous Euclidean Overlap Metric (HEOM) to overcome the limitations of distance-based modeling, making the computation of anomaly scores possible even in the presence of missing values. Also, the HEOM enables the anomaly detection algorithm to provide insights into the process by which it produces inferences at the sample level. The proposed algorithm's usefulness is validated through tests performed on several anomaly detection benchmarks from different domains and data modalities. Additional experiments on a non-proliferation use case are presented to show the suitability of the proposed anomaly detection method on data containing anomalies of different types and severity levels.

This paper is organized as follows. Section 2 introduces the proposed anomaly detector. The datasets used for benchmarking and validating the detector are described in Section 3. The experiments setup is described in Section 4. Section 5 presents the results and discussion, and the conclusion and future work are discussed in Section 6.

2. Anomaly Detection Algorithm

Based on the data and model transparency challenges described in the *Introduction*, we design a general anomaly detection algorithm (SOM+kNN) applicable to any ML learning problem to decide only upon its functioning instead of its execution. This allows for abstractions that enable anomaly detection model interpretability down the road.

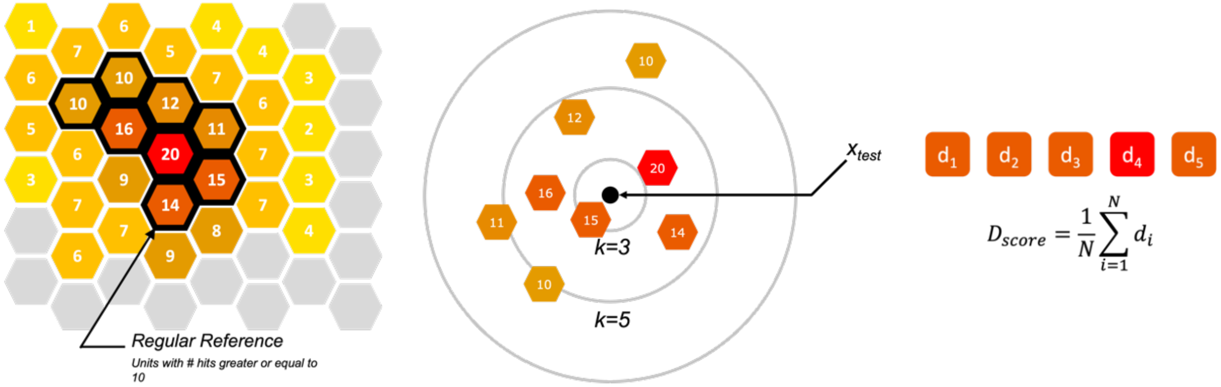
Regular Reference
Units with # hits greater or equal to 10
x_{test}
k=3
k=5
d_1 d_2 d_3 d_4 d_5
$D_{score} = \frac{1}{N}\sum_{i=1}^{N} d_i$

Figure 1. A realization of the anomaly detection algorithm second stage. The neurons with most of the samples associated with them to a degree (10) are extracted from the anomaly detection model. The distance between an unseen sample and its five nearest neighbors from the standard reference is computed. The mean is used to aggregate the distances to obtain the discrepancy score for the sample.

2.1 Step-by-step Anomaly Detection Algorithm

The input to the anomaly detection algorithm consists of tabular data containing the set of plausible scenarios from the case or system to be analyzed. The algorithm goes through a two-stage step-by-step process to (1) train the detector and compute the anomaly threshold parameter to subsequently (2) flag unseen samples as anomalous and non-anomalous. In the first stage during training, the samples in the train set are fed to the network of a SOM [9] model using competitive learning.

In the first stage, for each training sample, the distances to all the weights vectors of the network are computed using the HEOM [10] (See Figure 1). For each dimension, the HEOM outputs zero if both values are missing and one if only one is missing. If neither is missing, it outputs the absolute difference between the values divided by the overall range of values in that dimension. The arithmetic mean of the per-dimension distances then gives the vector distance and the weights of the best matching unit and the ones close to it are adjusted toward the sample. This process is repeated for several thousand iterations until the model converges.

The units with the highest number of training samples associated with them during training are extracted from the map of the SOM to build the standard reference. The reference represents the universe of expected scenarios; behavior significantly different from the standard reference should be considered anomalous. For each sample in the train set, the k nearest neighbors from the standard reference are found using the nearest neighbor algorithm [11], and the distance from them is computed. Taking the mean over the distances from the neighbors for each training sample yields their discrepancy scores. The three-sigma rule is then used to compute the anomaly threshold.

In the second stage, the discrepancy score is computed for samples the anomaly detection model has not seen during training. The scores are compared to the anomaly threshold obtained in the first stage. Scores over the threshold are deemed anomalous, whereas scores on or under the threshold are non-anomalous. Given that the HEOM computes discrepancy scores over the present and missing values individually, the proposed SOM+kNN becomes transparent in the context of the contribution of each type of distance to explain how it determines whether a sample is abnormal or not. The *Results and Discussion* section provides a discussion of the explanations for several inferences produced by the proposed anomaly detection algorithm.

3. Datasets

To evaluate the proposed anomaly detection algorithm, we source more than a dozen datasets from ADBench, an anomaly detection benchmark with a comprehensive set of methods, real-world and synthetic datasets, and capabilities for anomalous data generation [8]. Additionally, we validate the algorithm on an actual non-proliferation use case. The following is a description of the datasets and anomaly types considered in our experiments for evaluating and comparing our proposed algorithm to other unsupervised anomaly detection models.

3.1 ADBench

For the sake of fair comparison, the task of identifying irregular data samples is limited to the problem of unsupervised anomaly detection. In this problem, an anomaly detection algorithm AD is presented with a collection of n samples $\boldsymbol{X} = \{x, \ldots, x_n\} \in \mathbb{R}^{n \times d}$, where each sample contains f features. With this setting, the goal is to train a model M to output anomaly score $\boldsymbol{A_s} = M(\boldsymbol{X}) \in \mathbb{R}^{n \times 1}$, denoting the level of outlyingness for each sample. Subsequently, predictions are performed on m unseen samples $\boldsymbol{X_{test}} \in \mathbb{R}^{m \times d}$ to output $\boldsymbol{A_{s-test}} = M(\boldsymbol{X_{test}}) \in \mathbb{R}^{m \times 1}$. Note that unsupervised anomaly detection methods do not use data labels during training in this setting. The labels are only used for investigating model performance across different datasets, anomaly types, and severity levels post-training.

Table 1. Benchmark datasets from ADBench [8].

Dataset	Description	Number of Samples	Number of Features	Percentage of Anomalies	Category
Annthyroid	Medical information about hypothyroidism.	7200	6	7.42	Healthcare
Glass	Forensic data describing types of glass.	214	7	4.21	Forensic
Ionosphere	Radar data of electrons in the ionosphere showing evidence of some type of structure in the ionosphere.	351	33	35.90	Oryctognosy
MAGIC Gamma	Registration of high energy gamma particles.	19020	10	35.16	Physical
MuSK	Multivariate samples describing Muscle-specific kinase conformations.	3062	166	3.17	Chemistry
Shuttle	Aeronautic information about NASA space shuttles.	49097	9	7.15	Astronautics
Spambase	Spam and non-spam e-mails.	4207	57	39.91	Document
Waveform	Data representing three classes of waveforms.	3443	21	2.90	Physics
Wilt	Differentiates diseased trees from other land covers.	4819	5	5.33	Botany
Yeast	Cellular localization sites of proteins.	1484	8	34.16	Biology

We leverage the work from ADBench to generate a baseline of realistic synthetic datasets from ten diverse benchmarks (See Table 1. for a more detailed description).

3.2 Non-proliferation Use Case

In addition to the ADBench datasets, we validate the usefulness of the proposed anomaly detection algorithm on an actual non-proliferation use case containing a set of real-valued data streams. At each time step, statistical features such as quantile, mean, and standard deviation are extracted from the last few hours of each data stream; if no data is present in that period, the features are left missing. The anomalies inserted into the dataset are the following:

- *Feature removal*: This setting removes a subset of features across all time points following feature extraction.
- *Feature zeroing*: This setting zero out a specific subset of features across all time points following feature extraction.
- *Feature randomization*: This setting shuffles the feature values across its time points following feature extraction.

These anomalies are applied to 1, 4, 8, 10, and 12 features of the non-proliferation dataset.

4. Experiments

The study conducted on ADBench's benchmarks provides a standardized scaffold for evaluating the performance of different anomaly detection methods, which include other traditional and modern deep learning-based anomaly detection approaches. Table 2 lists the detectors considered in this work and briefly describes each and their capabilities in terms of explainability, handling missing values, and processing high-dimensional datasets. These methods and the proposed SOM+kNN algorithm are applied to data subjected to the corruptions and severity levels described in the *Datasets* section.

Table 2. Comparison of anomaly detection methods on their capabilities to handle missing and high-dimensional data and if they can explain the inferences they produce [8].

Model Name	Explainable	Missing Values	High-dimensional Data	Model Description
Cluster-Based Local Outlier Factor (CBLOF)	✓		✓	A density-based method that uses clustering to identify standard examples and calculates the local outlier factor for each sample based on its distance to the nearest cluster.
Local Outlier Factor (LOF)	✓		✓	A density-based outlier detection method calculates an outlier factor for each example based on the ratio of its local density compared to the densities of its nearest neighbors in the feature space.
Isolated Forests (iForest)	✓		✓	An ensemble of isolation trees is used for anomaly detection by considering the path length of each sample.
Connectivity-based Outlier Factor (COF)	✓		✓	A density-based method that calculates an outlier factor for each sample based on its distance to the nearest neighbors in the feature space and the connectivity of these neighbors.
Deep Autoencoding Gaussian Mixture Model (DA GMM)			✓	A deep learning method that combines autoencoding and Gaussian mixture modeling to identify anomalous examples.
Subspace Outlier Detection (SOD)			✓	An anomaly detector that considers the relationships between the features in a subspace and identifies anomalous examples.
Copula-Based Outlier Detection (COPOD)			✓	A distribution-based method that models the dependencies between the features using copulas and calculates the likelihood of each sample under the copula model.

Empirical-Cumulative-distribution-based Outlier Detection (ECOD)	✓		✓	A Distribution-based method that estimates a data's empirical cumulative distribution function (ECDF) and identifies samples outside a specific interval around the ECDF as anomalous.
k-Nearest Neighbors (kNN)			✓	An algorithm that identifies anomalies based on the classes of its k nearest samples using the Euclidean distance.
Histogram-based Outlier Score (HBOS)	✓		✓	A density-based method that uses histograms to estimate the density of the data and calculates the outlier score for each example based on its distance to the nearest bin in the histogram.
Principal Component Analysis (PCA)			✓	Dimensionality reduction method that projects the data onto a lower-dimensional subspace and identifies anomalous examples far away from the central cluster in the subspace.
Lightweight On-line Detector of Anomalies (LODA)			✓	Decision tree-based method that learns the data's ordinary behavior and identifies examples that deviate from the learned behavior as anomalous.
One-Class Support Vector Machines (OCSVM)			✓	A boundary-based method that uses an SVM to learn a boundary around ordinary examples and identifies anomalous samples falling outside the threshold.
Deep Support Vector Data Description (DeepSVDD)			✓	A deep learning method that uses an SVM-based model to learn a compact and tight representation of the standard examples and identify examples that fall outside the representation as anomalous.
Proposed SOM+kNN Algorithm	✓	✓	✓	**Combines the SOM, kNN, and HEOM for detecting anomalies in data with complex structures or high dimensionality and providing explainable inferences.**

These state-of-the-art anomaly detection methods represent a diverse range of approaches to anomaly detection, like density-based methods, clustering-based methods, distribution-based methods, ensemble methods, and deep learning methods. They differ from the proposed SOM+kNN algorithm in the specific techniques used to build a schema and identify anomalous examples. For example, calculating the local outlier factor, modeling the dependencies between features, estimating the empirical cumulative distribution function, combining the outputs of multiple base models, using histograms to estimate density, projecting the data onto a lower-dimensional subspace, learning the ordinary behavior of the data, learning a boundary around the standard samples, and learning compact representations of the regular samples.

To evaluate the performance of the anomaly detection methods, we generate data for different types of datasets such as time series, images, financial information, video, speech, and text. This variety allows us to assess the performance of the detectors on a wide range of scenarios and compare them to the proposed SOM+kNN algorithm. Second, we generate training and test data based on a nuclear non-proliferation use case, including data streams from remote edge sensors deployed at a working industrial site. This approach allows us to validate the proposed SOM+kNN algorithm on data representative of a real-world scenario.

The metrics for the performance evaluation include the Area Under the Curve (AUC), which represents the area under the Receiver Operating Characteristic (ROC) curve. This metric allows us to assess the trade-off between sensitivity and specificity at the chosen anomaly threshold. Additionally, other metrics considered in this work include the precision and recall for measuring the overall false-positive and false-negative rates and the F1-score that evaluates the overall recall and precision. Finally, the accuracy provides insights into the overall performance of detecting positive and negative samples correctly.

5. Results and Discussion

This section presents the results and discussion of applying the anomaly detection methods and the proposed SOM+kNN to the datasets generated using ADBench and the actual non-proliferation case.

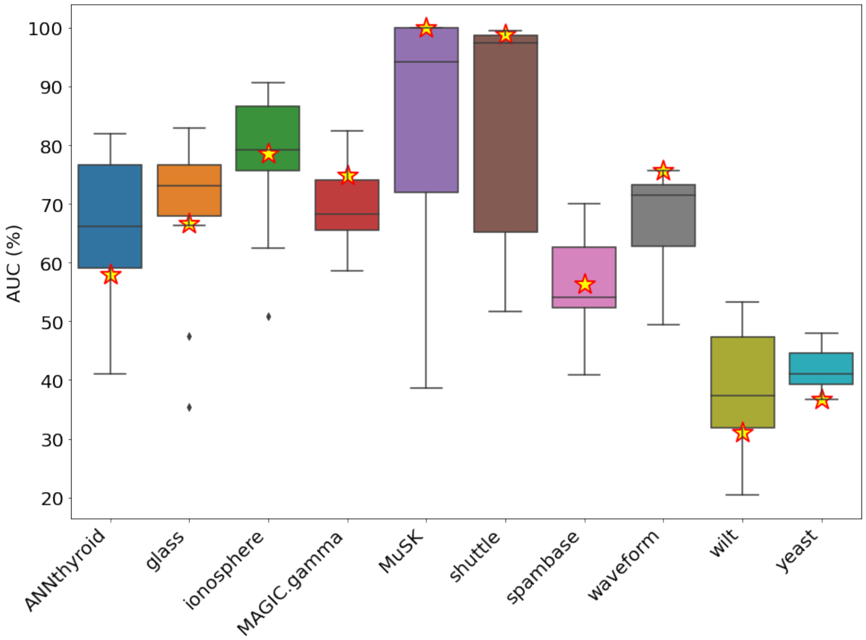

Figure 2. Comparison of several anomaly detection methods on their AUC performance. The performance obtained by the proposed SOM+kNN algorithm is highlighted using the star icon.

5.1 ADBench

Figure 2 depicts the performance obtained by the proposed anomaly algorithm highlighted with the star icon and the performances obtained from the baseline anomaly detection models from ADBench. The SOM+kNN algorithm obtains better or comparable AUC scores to those of the other methods over the Ionosphere, MACIG.gamma, MuSK, Shuttle, Spambase, and Waveform datasets. This suggests a proportional relationship between the performance of the SOM+kNN and the number of samples and features in the datasets.

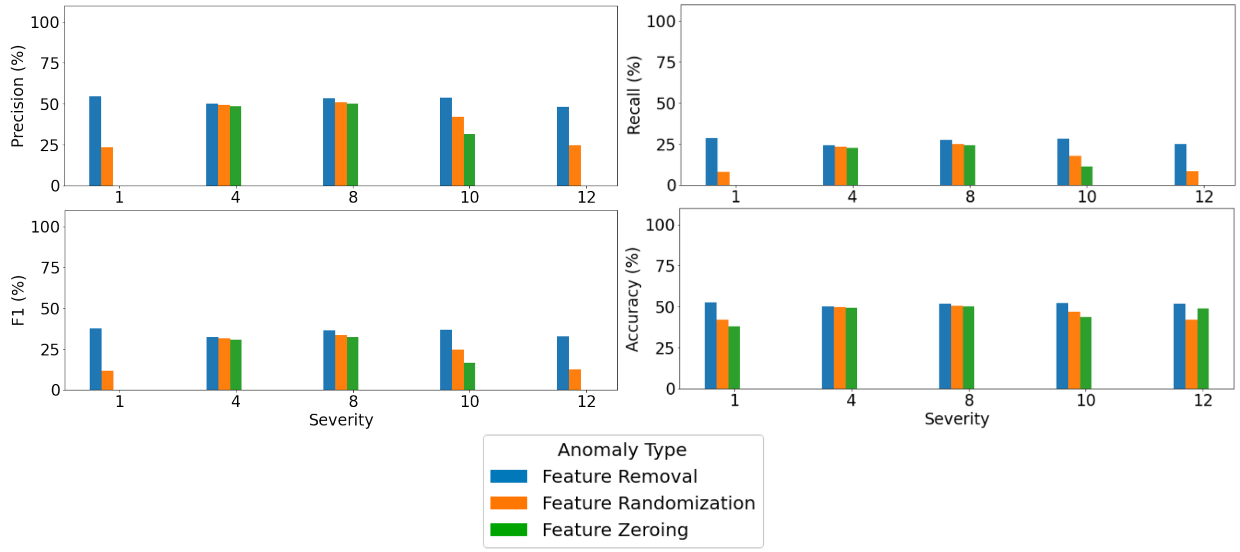

Figure 3. Performance results for the proposed SOM+kNN algorithm on the non-proliferation use case.

5.2 Non-proliferation Use Case

Figure 3 shows that the SOM+kNN obtains values for the precision, recall, F1, and accuracy metrics of around 50%, 25%, 30%, and 50%, respectively, across all the severity levels. On average, the model performs better when the severity level is equal to 4 and 8. This means that the model is better at detecting anomalous samples when there are several features affected by the corruptions inserted into them. Regarding the missing values present in the data, we hypothesize that they contribute to most of the discrepancy scores producing false positive predictions. This is confirmed by the results in Figure 4, depicting the individual contribution of present and missing values of four samples over different anomaly types and the threshold. The samples marked as false positives over all the anomaly types show that the missing values are the driving factor of the discrepancy score in producing a one when it should be a zero. This could be an indication of data sparsity from the edge sensors.

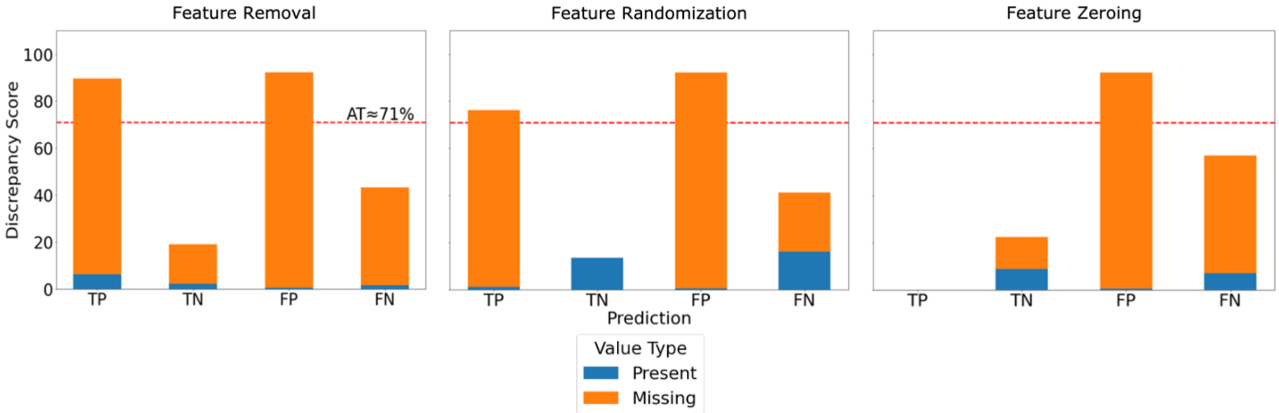

Figure 4. The individual contribution of present and missing values to the discrepancy score over four anomaly types at severity level equals to eight. The red dashed line represents the anomaly threshold.

6. Conclusion and Future Work

This work discussed the challenges associated with state-of-the-art anomaly detection approaches. These limitations include the lack of detectors that can simultaneously handle high-dimensional data and missing values while providing explanations for the inferences they produce. To alleviate these challenges, we presented the SOM+kNN anomaly detection algorithm, which combines the SOM, kNN, and HEOM to flag anomalies based on their discrepancy scores. Experiments on several real-world datasets demonstrated the proposed anomaly detection algorithm's usefulness in identifying anomalies. Additional experiments on a non-proliferation use case validated the algorithm to detect anomalies and explain different types of predictions at the sample level using the individual present and missing values discrepancy scores. Moreover, the algorithm is suitable for non-proliferation applications, given its verified repeatability regarding the results obtained for different datasets. Future work includes transferring knowledge from the current use case to other testbeds and domains.

7. Acknowledgements

We are grateful for the support of the National Nuclear Security Administration, Office of Defense Nuclear Nonproliferation towards this research under the Persistent DyNAMICS venture. A special thank you to the Office Program Manager Christopher Ramos for all their wisdom and guidance to make this presentation and venture a success.

References

[1] Kitcher, Evans D., Jeremy M. Osborn, and Sunil S. Chirayath. "Characterization of plutonium for nuclear forensics using machine learning techniques." *Annals of Nuclear Energy* 170 (2022): 108987.

[2] Zhu, Xiao Xiang, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. "Deep learning in remote sensing: A comprehensive review and list of resources." *IEEE geoscience and remote sensing magazine* 5, no. 4 (2017): 8-36.

[3] Surapaneni, Ravi Kishan, Sailaja Nimmagadda, and Roja Rani Govada. "Handling Incomplete and Delayed Information Using Optimal Scheduling of Big Data Stream." In *International Conference on Intelligent Computing and Smart Communication 2019: Proceedings of ICSC 2019*, pp. 147-157. Springer Singapore, 2020.

[4] Nassif, Ali Bou, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. "Machine learning for anomaly detection: A systematic review." IEEE Access 9 (2021): 78658-78700.

[5] Zheng, Jian, Jingyi Li, Cong Liu, Jianfeng Wang, Jiang Li, and Hongling Liu. "Anomaly detection for high-dimensional space using deep hypersphere fused with probability approach." *Complex & Intelligent Systems* 8, no. 5 (2022): 4205-4220.

[6] Zemicheal, Tadesse, and Thomas G. Dietterich. "Anomaly detection in the presence of missing values for weather data quality control." In *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 65-73. 2019.

[7] Li, Zhong, Yuxuan Zhu, and Matthijs van Leeuwen. "A Survey on Explainable Anomaly Detection." *arXiv preprint arXiv:2210.06959* (2022).

[8] Han, Songqiao, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. "Adbench: Anomaly detection benchmark." *Advances in Neural Information Processing Systems* 35 (2022): 32142-32159.

[9] Kohonen, Teuvo. "The self-organizing map." *Proceedings of the IEEE* 78, no. 9 (1990): 1464-1480.

[10] Wilson, D. Randall, and Tony R. Martinez. "Improved heterogeneous distance functions." *Journal of artificial intelligence research* 6 (1997): 1-34.

[11] Fix, Evelyn. *Discriminatory analysis: nonparametric discrimination, consistency properties*. Vol. 1. USAF school of Aviation Medicine, 1985.