COVERT COGNIZANCE: EMBEDDED INTELLIGENCE FOR INDUSTRIAL SYSTEMS

by

Arvind Sundaram

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Nuclear Engineering

West Lafayette, Indiana

December 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Hany Abdel-Khalik, Chair
School of Nuclear Engineering

Dr. Berkay Celik
Department of Computer Science

Dr. Stylianos Chatzidakis
School of Nuclear Engineering

Dr. Alexander Hagen
Pacific Northwest National Laboratory

Dr. Lefteri Tsoukalas
School of Nuclear Engineering

Approved by:
Dr. Seungjin Kim

Dedicated to my aunt Meena, the greatest teacher I knew

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Hany Abdel-Khalik. From the first pep-talk you gave me in my junior year in 2018 to this day, you have never ceased to amaze me with your work ethic, discipline, knowledge, and passion. The phrase "think integral not differential" is something I'm definitely going to steal from you in the hope that it inspires others like it inspired me. I will miss the late-night chance encounters at FLEX and our digressions about current affairs and our life endeavors. You have been more than just an advisor to me; you have been the older brother I never had, and I don't think one paragraph can do justice to the role you have played in my life and the ambition you have instilled in me.

I am grateful to my graduate committee members – Dr. Celik, Dr. Chatzidakis, Dr. Hagen, and Dr. Tsoukalas for their invaluable guidance and advice on my dissertation topic. The value of a third set of eyes is often understated, and I really appreciate the new perspectives you have offered.

I would like to thank my colleagues, Jeongwon Seo, Dr. Li, and Dr. Huang, who are both excellent mentors and friends. The countless hours spent learning, traveling, ranting, eating, and working out together made my Ph.D. experience much more enjoyable, and I thoroughly enjoyed the amalgamation of our vastly different cultures. A special mention goes out to my friends Haoxuan Wang, Tyler Ray, and Arjun Mannem, to whom I am grateful for keeping me cheerful and in good mental health during stressful times.

I believe this section would be incomplete without mentioning the people that have sacrificed the most to make this dissertation a reality – my family. My success would not have possible without my parents, their unwavering support and belief in me, their assistance emotionally and financially through the years, and their personal sacrifices to ensure that my education was a top priority. Additionally, I would like to thank my extended family for making me feel at home here in the U.S. They were a second set of parents to me and ensured that I had a place to go if I ever felt homesick. I am truly grateful to be blessed with a loving family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
APT	:	Advanced Persistent Threat
C2	:	Covert Cognizance
CNN	:	Convolutional Neural Network
CPS	:	Cyber-physical systems
DCPM	:	Direct Current Permanent Magnet
DIOD	:	Deceptive Infusion of Data
DOF	:	Degree Of Freedom
DNN	:	Deep Neural Network
FDIA	:	False Data Injection Attack
FN	:	False Negative
FP	:	False Positive
GAN	:	Generative Adversarial Network
GAN-AD	:	Generative Adversarial Networks-based Anomaly Detection
GPU	:	Graphical Processing Unit
HMI	:	Human-Machine Interface
HOC	:	Higher-Order Component
IT	:	Information Technology
LOC	:	Lower-order component
LSTM	:	Long Short-Term Memory
LTI	:	Linear Time-Invariant
MAD-GAN	:	Multivariate Anomaly Detection with Generative Adversarial Networks
ML	:	Machine Learning
OT	:	Operational Technology
PCA	:	Principal Component Analysis
PERA	:	Purdue Enterprise Reference Architecture
PLC	:	Programmable Logic Controller
PWR	:	Pressurized Water Reactor
ReLU	:	Rectified Linear Unit
RL	:	Reinforcement Learning
ROM	:	Reduced-Order Model(ing)
RPM	:	Revolutions Per Minute
SCADA	:	Supervisory Control And Data Acquisition
SMR	:	Small Modular Reactor
SVD	:	Singular Value Decomposition
SVM	:	Support Vector Machine
TN	:	True Negative
TP	:	True Positive

ABSTRACT

Can a critical industrial system, such as a nuclear reactor, be made self-aware and cognizant of its operational history? Can it alert authorities covertly to malicious intrusion without exposing its defense mechanisms? What if the intruders are highly knowledgeable adversaries, or even insiders that may have designed the system? This thesis addresses these research questions through a novel physical process defense called Covert Cognizance (C2).

C2 serves as a last line of defense to industrial systems when existing information and operational technology defenses have been breached by advanced persistent threat (APT) actors or insiders. It is an active form of defense that may be embedded in an existing system to induce intelligence, i.e., self-awareness, and make various subsystems aware of each other. It interacts with the system at the process level and provides an additional layer of security to the process data therein without the need of a human in the loop.

The C2 paradigm is founded on two core requirements – zero-impact and zero-observability. Departing from contemporary active defenses, zero-impact requires a successful implementation to leave no footprint on the system ensuring identical operation while zero-observability requires that the embedding is immune to pattern-discovery algorithms. In other words, a third-party such as a malicious intruder must be unable to detect the presence of the C2 defense based on observation of the process data, even when augmented by machine learning tools that are adept at pattern discovery.

In the present work, nuclear reactor simulations are embedded with the C2 defense to induce awareness across subsystems and defend them against highly knowledgeable adversaries that have bypassed existing safeguards such as model-based defenses. Specifically, the subsystems are made aware of each other by embedding critical information from the process variables of one sub-module along the noise of the process variables of another, thus rendering the implementation covert and immune to pattern discovery. The implementation is validated using generative adversarial nets, representing a state-of-the-art machine learning tool, and statistical analysis of the reactor states, control inputs, outputs etc. The work is also extended to data masking applications via the deceptive infusion of data (DIOD) paradigm. Future work focuses on the development of automated C2 modules for “plug ‘n’ play” deployment onto critical infrastructure and/or their digital twins.

1. ARTIFICIAL INTELLIGENCE

Assume a drone is flown into unfriendly territory and captured. Experts may attempt to reverse-engineer the drone to understand and uncover hidden relationships among its data and other valuable information. Is it possible to make the drone intelligent enough to recognize the situation based on how it is being operated? Is it possible to feed false information to mislead the captors covertly? These are the key research questions explored in this chapter.

The definition of intelligence varies from researcher to researcher, but in the context of the present work, the definition of artificial intelligence (AI) by Albus in [1], "The ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal," is quite befitting. In other words, an intelligent drone is aware of its execution history, and autonomously takes actions that maximize its chances of success, i.e., to prevent reverse-engineering, without the need for external input or supervision. The million-dollar question is: how?

1.1 History of AI

The roots of AI research may be traced to classical philosophers and the invention of the digital computer in the 1940s. It was believed that intelligence and thought could be formalized through mathematical manipulation and logic, with conjectures such as the Church-Turing hypothesis [2] providing the breakthrough necessary to formulate mathematical logic through binary operations. The goal was to emulate the human brain electronically and mimic synapses and electrical pulses through artificial neurons and activation functions. With the advancements in digital computing, researchers quickly wrote programs for checkers and chess and solving mathematical theorems, culminating in the famous Dartmouth Workshop of 1956 that led to the coining of the term AI.

AI was no exception to the famous Gartner hype cycle [3]. The following decades were highly optimistic with major funding of AI research and predictions of fully intelligent or "strong AI" agents within a generation. However, by the 1970s, data storage and computational power limitations, lack of progress in vision and robotics, inadequacy of the neuron model, and the failure

of commercialization ventures led to an "AI winter" until the 1990s. Advancements in chip technology and mass storage capabilities in the 1990s mitigated some of the problems associated with AI, specifically the subset known as machine learning (ML) that attempts to "learn by example" from vast amounts of data.

AI/ML research has undergone a monumental resurgence since the 1990s finding vast applications in imaging, engineering, biology [4]–[11] etc. through the development of machine learning algorithms such as support vector machines (SVMs) [12] and the introduction of deep neural networks (DNNs) [13]. The 2010s saw a new class of learning algorithms called generative adversarial nets (GANs) [14] that allowed neural nets to compete against each other in a zero-sum game until one net, called the generator, can mimic the training dataset successfully without being detected as fake by the other, called the discriminator. This development led to the creation of photorealistic images, colloquially known as "deepfakes" [15], and has found uses in AI art to create portraits, melodies etc. In parallel, artificial general intelligence [16] continues to be explored in the hopes of realizing the original goals of AI such as cognition and fully intelligent systems.

A discussion on AI would be incomplete without its effects on humanity. Recent years have overseen an interest in explainable AI in attempt to explain the actions and decision-making process of an intelligent agent in human terms. As AI enjoys success in facial recognition, object recognition, and autonomous control applications, further analysis and discussion is warranted on the inner working, modes of failure etc. of machine learning algorithms to ensure that they are used to the benefit of humanity. Greater transparency and accountability, and the need to respect data privacy is of paramount importance to ensure sustainable development of AI algorithms [17]. Researchers have alerted to the existence of algorithmic bias in facial and voice recognition systems, hiring practices, and raised several questions regarding the liability of autonomous agents such as self-driving cars [18]–[20]. Additionally, due to the infancy of explainable AI, it is exceedingly difficult to judge whether AI/ML algorithms are impartial, fair, and trustworthy in their judgment. The rising popularity of deepfakes has also raised several ethical and security questions due to their realistic appearance and potential for misuse. In response, various organizations such as the European Commission, the OECD, and the US government have released several directives guiding and regulating the use of AI in modern society [21], [22]. The following section delves into some of these major applications of AI in the modern world.

1.2 Applications of AI

AI has found applications in fields such as astronomy, finance, gaming, healthcare, industry, transportation etc. In the finance industry, AI/ML algorithms have been leveraged to predict stock movements, perform risk analysis, and build automated trading algorithms [23]. In gaming, GANs have been used to "upscale" graphics, i.e., make old video game graphics look modern [24]. Other prominent advances in AI/ML research include the development AlphaGo [25] in Weiqi/Go and OpenAI [26] in Dota 2 that proceeded to defeat the top-ranked player and team respectively. Both AI bots were developed using reinforcement learning (RL) and trained by playing hundreds of thousands of games with itself. Additionally, GANs have been used to create AI art by training on vast amounts of portrait images and generating a new portrait based on information gleaned from the training data [27]. They have also been used for humorous, malicious, and advertising purposes by creating deepfake videos of politicians and celebrities giving speeches, most famously that of former US President Barack Obama [28].

However, the focus of the present manuscript is on industrial applications. With the advent of digitalization and "big data" storage capabilities, industrial systems have been one of the largest beneficiaries of AI technology. Recent years have overseen a paradigm shift in industrial infrastructure through the cyber-physical framework [29], resulting in a large-scale integration of sensors, controllers, and other physical systems with computers to form cyber-physical systems (CPS). With the help of cyber technology, so-called "smart" devices have been developed to further optimize resource allocation, perform predictive analytics, detect intrusion etc. and adapt to the need of the hour. The "brain" of these devices, also called predictive modeling technology, has proven to be effective at behavior forecasting, surveillance, and control strategy and requires the underlying model to be continuously updated to keep pace with the functional requirements of the CPS. The underlying model is typically constructed based on observations of the physical system using a combination of domain expertise and data-driven ML algorithms.

A commonly encountered task in CPS closely related to classification is that of anomaly detection and condition monitoring. These tasks may be supervised, unsupervised, or semi-supervised. In supervised tasks, the data is prelabeled with the target class and the goal of the algorithm is to minimize the misclassification error. In unsupervised tasks such as clustering, there is no information on the class label and algorithms attempt to cluster the data into classes based on a measure of separation such as Euclidean distance. In semi-supervised problems, only a small set

of instances are labeled, and the goal of the ML tool is to generalize to a wider set. For example, ML has been used in the plastic industry to detect low-quality production cycles in a production line of coffee capsules [30]. Five predictive models, namely, k-nearest neighbor, naïve Bayes, random forest, decision trees and support vector machines, were trained and tested on the MONSOON Coffee Capsule dataset to detect defective cycles. F1 scores of 0.657 and accuracies of close to 70% were reported using the k-nearest-neighbors algorithm, leading to a significant reduction in waste. Similar research has been done using data from a reflow oven to perform predictive maintenance. For highly specific environments heavily dependent on domain knowledge, random forest algorithms have been shown to perform better than other ML algorithms such as SVMs and DNNs.

In the nuclear industry, model-based defenses have been constructed by exploiting physical correlations between sensors to extract signatures from time-series data [31]. The signatures, depending on their intensity, may be classified as lower-order or higher-order components (LOCs and HOCs), with lower-order components representing dominant trends and higher-order components representing subtle and more noise-dominated trends. SVMs may be employed to draw a decision boundary to separate anomalous data from normal data using these LOCs and HOCs, and additionally, the anomalous data may be isolated to certain fault categories based on further classification. While LOCs are often adequate to detect large deviations from normal behavior, they are augmented with HOCs to detect more subtle deviations such as sensor drift and incipient signs of equipment degradation that occur over longer timescales. With regards to materials, convolutional neural networks (CNNs) have been developed to analyze small-scale images of structural components for crack and corrosion detection [32]. These may be helpful in detecting incipient signs of damage in an automated fashion by scanning the surface of structural components and passing the images to a trained neural network for classification.

ML algorithms such as neural networks and RL have also been used for optimization of parameters in various industrial processes such as press hardening, production planning, and quality assurance [33]–[35]. For example, principles from least-squares regression and function optimization were used for quality assessment and to adjust temperature, spacing, and force parameters for optimal press hardening. In the nuclear industry, the DOE has funded a project to automate the additive manufacturing (3D printing) of microreactors using an intelligent agent

guided by a RL framework to perform quality assurance and adjust the printing process accordingly.

Another common use of ML algorithms such as principal component analysis (PCA) is to build reduced-order models (ROMs) for industrial systems [36]–[40]. PCA is an unsupervised technique that provides a linearly uncorrelated basis ranked by total variance explained. It helps isolate features of the data and plays a major role in denoising applications, exploratory data analysis etc. The constructed ROMs may serve as surrogate models to save computational resources, provide a simple low-dimensional representation of the data/system, identify dominant patterns etc. They often find use in simulations of complex systems such as airflow and reactor simulations with millions of variables that can be reduced to just few tens of variables that carry most of the variance within the data. Additionally, PCA-based algorithms offer a great deal of flexibility since they allow users to control the error in the data by prespecifying a tolerance, and only keeping as many principal components as necessary.

For example, in the nuclear industry, ROMs serve as a useful tool for uncertainty quantification and model validation [41]–[44]. There is a strong incentive to account for uncertainties in the critical eigenvalue of reactors since a tighter bound on the uncertainty may allow for the reactor to operate at potentially more efficient conditions. These uncertainties are a typically the result of uncertainties in the nuclear cross-section data (input) that propagate through reactor calculations and affect the final estimate of the eigenvalue of the reactor (output). Gaining an accurate understanding of the various uncertainties in the reactor, however, requires several executions which are often intractable for complex high-fidelity 3D models and require high performance computing even with accelerated solver methods such as Krylov subspace methods, method of characteristics etc. This is typically due to the vast number of energy groups in neutronics calculations coupled with the fine mesh requirements of thermal hydraulics calculations. Using ROM-based input-output surrogate models, this computational burden is greatly reduced facilitating quick execution of an approximate surrogate model and providing an upper bound on the error due to the reduction, which may be accounted for in the overall uncertainty of the eigenvalue. However, it is noted that ROMs often require an upfront one-time investment to construct the model based on the high-fidelity 3D simulations, typically done using randomized linear algebra techniques [45], [46]. These techniques find the active subspace of the model that describes most of the variance through random perturbations of the model inputs.

With regards to security, the pattern detection nature of ML algorithms has been successfully leveraged to detect zero-day vulnerabilities in hardware using deep learning, network intrusion via packet analysis, and data tampering via sensor fingerprinting [47]–[49]. ML algorithms have been deployed to develop so-called model-based defenses that rely on observing the data and detecting anomalies based on statistical discrepancies with a reference model that may be data-driven and/or physics model-based [50]. For instance, sensors at a water distribution facility were fingerprinted by tying the noise in the data to a particular sensor, rendering it difficult to fake data [49]. If noise analysis of the data did not bear the statistical properties of the sensor (the "fingerprint"), it was determined to be anomalous. In the nuclear community, multilayer cyber-attack detection systems have been developed to detect intrusion based on network traffic data, host system data, and finally, the process data itself [36]. A common shortcoming of these model-based defenses is that they may be bypassed by a knowledgeable adversary. The physics of critical infrastructure such as nuclear powerplants is well-known and can be found in most introductory textbooks, thus weakening the assumption that an adversary may not have domain knowledge. With the advent of physics-informed neural networks in recent years, it may now be possible to find critical parameters by formulating an inverse problem and training the network with vast amounts of industrial data. In fact, research has shown that AI may be used to learn fingerprints and the underlying physical model using the same data-driven ML algorithms used to create the fingerprint [39], akin to counterfeiting money using authentic but unauthorized plates. Tools such as SINDy and PDE-FIND may be utilized to uncover underlying differential equations and physics of data from unknown sources, greatly simplifying the problem for an adversary without domain knowledge [51], [52]. The adversary may simply train surrogate models to the desired level of accuracy after an initial lie-in period to collect data.

Returning to the example of the intelligent drone at the beginning of this chapter, it is evident that misleading its captors is a complex challenge that must withstand reverse-engineering efforts by both humans and AI. Despite being adept at pattern discovery, AI and ML algorithms have shortcomings and blind spots that may be exploited, as outlined in the following section.

1.3 Shortcomings of AI/ML

This section provides a comprehensive literature review on some AI/ML algorithms such as DNNs, SVMs etc and their shortcomings. DNNs are a popular choice of ML tools that have

enjoyed mainstream success across many fields such as image processing, time-series forecasting, anomaly detection, object identification etc. It is a neural network consisting of multiple layers of neurons, each with its own weights, biases, and activation functions. These networks are a practical implementation of the universal approximation theorem, which states that a simple feedforward neural network with a single hidden layer of infinite size can approximate any continuous function. In the case of DNNs, the size of the layer is finite but multiple layers are stacked to increase the width of the network and achieve a similar effect, although they may not necessarily be universal approximators depending on the dimension of the input layer. They are often trained using gradient-based backpropagation algorithms that adjust the weights in every iteration depending on the error in the output. The training is typically formulated as an optimization problem with differentiable loss functions such as mean-squared error that must be minimized through adjustment of the neuronal weights and biases. Consequently, due to the vast number of neuronal parameters that must be estimated, DNNs are often data-hungry and require several hours of training even on modern graphical processing units (GPUs).

One of the common pitfalls of training DNNs is the vanishing gradient problem [53], where the backpropagation algorithm is unable to adjust the weights due to a zero gradient at a local minimum. Moreover, increasing the complexity of the network with multiple layers and/or neurons often increases the odds of encountering a local minimum since the gradient is computed using the chain rule in calculus which involves multiplication of multiple fractions resulting in very small adjustments to the weights. Recent years have seen the introduction of residual networks (ResNets) [6] with skip connections to alleviate this issue. These skip connections allow information to connect a layer to a deeper layer without necessarily passing through intermediate layers, thus allowing information to permeate through the network, effectively reducing network complexity and mitigating the impact of vanishing gradients during the initial training phase. As further training occurs, skipping is reduced and the intermediate layers are trained with the expectation that the weights of the network are closer to their optimal values. Other solutions to combat vanishing gradients include using the rectifier activation function (ReLU), long short-term memory networks, and using a search algorithm for weight optimization instead of backpropagation to train the network.

Secondly, DNNs often suffer from a lack of interpretability due to its structure. Constructing DNN architectures is an art, and often relies on trial-and-error and rule-of-thumb

guidelines with little reasoning. Oftentimes, existing architectures that "work" are modified and adapted to the target task with several regularization techniques to guide the training such as dropout layers that may randomly deactivate during training to avoid overfitting. Additionally, intermediate outputs from each network's layer are often incomprehensible to humans, leading to DNNs having the appearance of a "black box". For instance, DNNs used in the classification of animals may show features such as eyes, nose, face, fur etc. being extracted from the first layer, but the explainability is typically lost in subsequent layers even if the network classifies correctly. As a result, several questions arise as to what features the DNN learns, why some DNNs work well on certain datasets but not others, what the modes of failure are, etc. Understanding the inner working of these networks and why they succeed/fail is crucial to adoption in critical infrastructure applications. One popular yet controversial theory to explain the working of DNNs is the information bottleneck theory proposed in [54] as a foundational theory for deep learning. It attempts to explain deep learning using information compression and mutual information, where each layer attempts to compress the information in the signal of the previous layer while retaining as much relevant information as possible by eliminating some of the noise and signal in the compression. This departs from existing black box approaches that portray learning as an optimization problem alone where the goal is to minimize the error between the output and target variables. However, controversy stemmed from the fact that attempts to replicate the experiment with simple problems did not yield favorable results, with countercriticism being aimed at the methods used to compute mutual information in the replication experiments. Nevertheless, the theory is an important step in attempting to explain deep learning and may benefit from advancements in computational methods to estimate mutual information in high dimensions, such as the MINE algorithm.

SVMs are another class of supervised ML algorithms that perform well with high-dimensional separable data and may provide more interpretable results. The goal of these algorithms is to draw a decision boundary that ensures maximal separation between two or more class labels based on their features. If the data cannot be separated linearly in lower-dimensions, kernel functions such as the radial basis function project low-dimensional data onto a higher dimensional space to draw a hyperplane separating the datasets in the higher dimension (popularly known as the "kernel trick") [55], [56]. These methods are also robust to outliers since they rely on only a few critical "support" data points to determine the decision boundary. Nevertheless, they

often fail when encountering classes with overlapping that are not easily separable and require careful tuning of hyperparameters and the kernel function used until the desired performance is achieved. Additionally, they work best with smaller datasets and the training time often grows exponentially with data, thus rendering them unsuitable for big data applications without some preprocessing.

Naïve Bayes [56] is an ML algorithm that alleviates the issue of scalability and may run in real-time with multi-class classification capabilities. It utilizes Bayes theorem to estimate the posterior probability of an event and relies on two key assumptions – training data is representative of the population, and independence of the extracted features. This often renders it unsuitable for application in anomaly detection scenarios where anomalous data is often scarce and the features are typically correlated in time-series and/or image data. The probability estimates provided by the algorithm may not necessarily be meaningful outside a qualitative analysis due to violations of the underlying assumptions.

Decision trees [56], [57] are a simple yet powerful non-parametric classification tool in ML that models the classification problem using a series of simple decision rules inferred from the data features. These rules may be simple threshold-based decisions and are thus highly explainable in contrast to their neural network counterparts. Ensembles of decision trees may be constructed to solve complex problems; however, these structures often suffer from overfitting, high training times, and increased sensitivity to outliers. Random forests mitigate these challenges by minimizing the error of individual trees and forming a collective decision based on all trees. However, this comes at the cost of explainability as they often appear like black boxes in their working, which is often counterproductive to the inherent interpretability of decision trees.

Another example of a simple and interpretable ML tool is that of the k-nearest-neighbors algorithm [58]. Widely used for clustering, the algorithm relies on metrics such as Euclidean distance to determine class membership with a given data point assigned the most common class among its k-nearest-neighbors where k is typically a small integer. These algorithms are easy to interpret visually due to their reliance on geometrical metrics in lower dimensions and are ideal for unsupervised learning tasks to gain an initial idea of what data points may belong to the same class. However, the result is often not unique and multiple sets of clustering may occur for the same dataset requiring some additional domain knowledge for better performance. A common weakness of these algorithms is the curse of dimensionality, and they often require some

preprocessing such as PCA to reduce the dimension of the feature space. This is because the Euclidean distance metric does not generalize well to higher dimensions and many points may be equidistant from a given point. Additionally, the algorithm is extremely sensitive to scaling since it relies on distance-based metrics for clustering. It generally does not work well with imbalanced datasets and the algorithm typically does not scale well with large datasets.

A common shortcoming to most AI/ML algorithms is the presence of noise in the dataset. While noise may be introduced to improve the robustness of the training algorithm in some applications such as image classification, it often degrades the performances of AI/ML algorithms due to blurring of the decision boundaries by causing class overlap and increasing misclassification. Furthermore, adversarial examples [59] have been generated via the fast gradient sign method where seemingly noisy and nearly imperceptible perturbations to images cause DNNs to completely misclassify the object in ways a human would not, revealing the existence of a blind spot for AI/ML algorithms. Referring to the example of the captured drone, it appears that the insertion of noise may be a promising avenue of research to blind and evade AI/ML-based pattern discovery tools when model-based methods are bypassed. This forms the premise of the following section that delves into the efficacy of noise-based defense techniques against AI.

1.4 Noise: Can it blind AI?

The work presented in this section borrows from content previously published by the author in the journal *Progress in Nuclear* Energy in assessing the efficacy of operational technology (OT) active defenses [60]. Contrasting with model-based defenses that are passive since they rely on capturing deviations of the data and referencing it with a physical model/digital twin, active defenses involve interacting with the system through carefully tailored perturbations of the process variables. A well-known example in the CPS community is that of dynamic watermarking [61], where noise-like perturbations based on a hidden Markov model were inserted into the actuator of a control system. The system is then equipped with a χ^2-detector which computes the residual between the output of the system and the expected output. If a statistical analysis of the residual does not exhibit any trace of the perturbation, the data is deemed to be fake, and an alarm is raised. Another active algorithm that works on a similar premise is that of noise impulse integration where Gaussian noise is inserted and later removed from a control signal to detect tampering. A common shortcoming of these active defenses, however, is that they often

affect the optimality of the system due to the perturbations. For example, in dynamic watermarking, there is a tradeoff between the controller cost and the anomaly detection rate. While these techniques rely on the insertion of noise to secure the system, they are overt in that their presence can easily be detected by classification algorithms. In the preliminary assessment of such noise-based active defense techniques, it is argued that while noise may be used to carry information, albeit obfuscated, to secure CPS, it is crucial that these techniques remain undetected by ML algorithms by avoiding the use of patterns such as the same random seed or inducing correlations among the noise components. This is demonstrated through a series of numerical experiments where supervised and unsupervised learning algorithms are used to distinguish various types of noise.

In the first numerical experiment, supervised learning is used to demonstrate the capability of ML algorithms to identify different colors of noise based on raw data alone without the need for additional domain knowledge. While the raw time-series appears statistically random, the profiles differ in power spectrum, thus having markedly different patterns from Gaussian white noise. Additionally, it is also demonstrated that if the adversary possesses some domain knowledge, Fourier transformation of the raw time series to the frequency domain provides a clear linear separation between the different colors as shown in Figure 1 [62], allowing for easier classification. However, it is also observed that the classifier performance degrades between noise of similar colors (violet and azure) due to the relatively lower separability between the two classes compared to the other colors. Therefore, it is concluded that randomness alone is insufficient, and the statistical properties of noise must also be preserved in noise-based active defenses to evade detection by AI/ML algorithms.

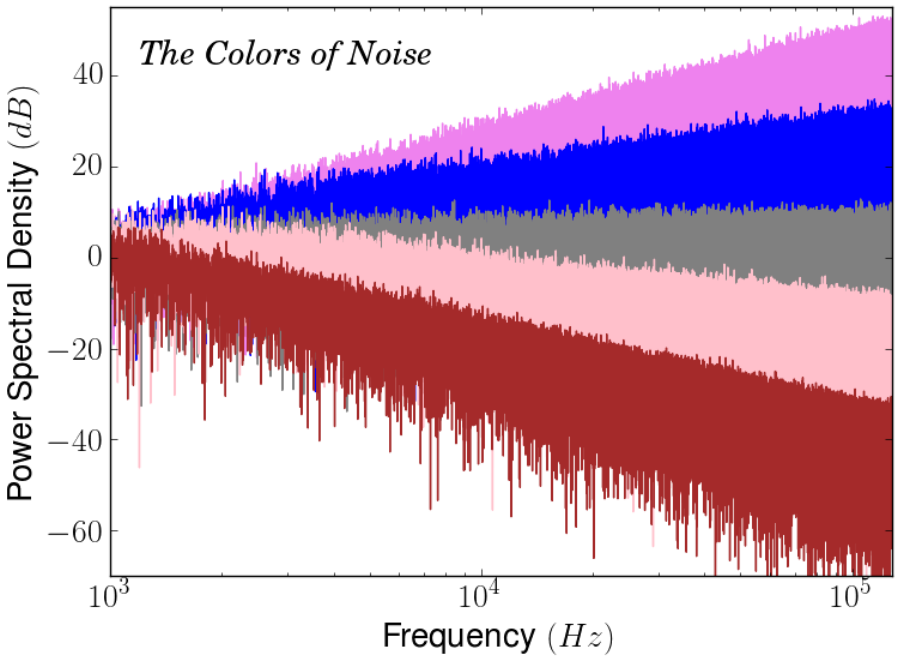

Figure 1: Colors of noise (Frequency domain)

In the second numerical experiment, the concept of inserting coded noise into process variables using secret keys is explored. The code here represents the seed of each random number generator used, which is then combined to generate noisy versions of the data. The KEYMASTER™ Generic Combined Cycle Gas Turbine Simulator System by WSC, Inc. (Figure 2) is simulated under normal operating conditions and the resultant dataset was corrupted with white Gaussian noise with a fraction of the datasets corrupted with the coded white Gaussian noise. While the two datasets are statistically similar in their noise profile, the noise in the coded datasets is generated using a linear combination of a few fixed random sources while the uncoded datasets had constantly changing random sources. It is observed that a simple feedforward neural network algorithm can detect the presence of fixed sources and classify the datasets into the coded and uncoded versions with complete accuracy. The detection capability persists even if the coefficients of the fixed sources are randomized with every run. However, the performance of the classifier degrades when more fixed sources are added, but this is alleviated with additional training samples as shown in Figure 3 below.

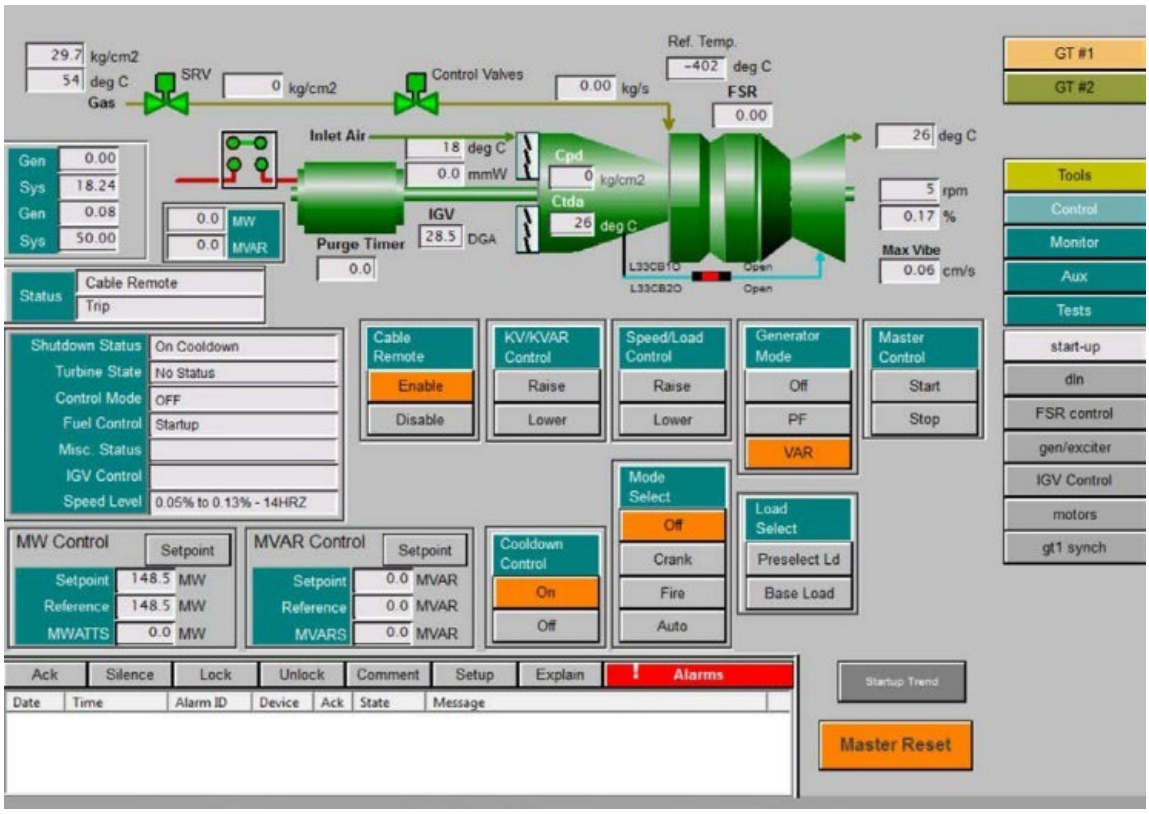

Figure 2: WSC Gas Turbine Simulator

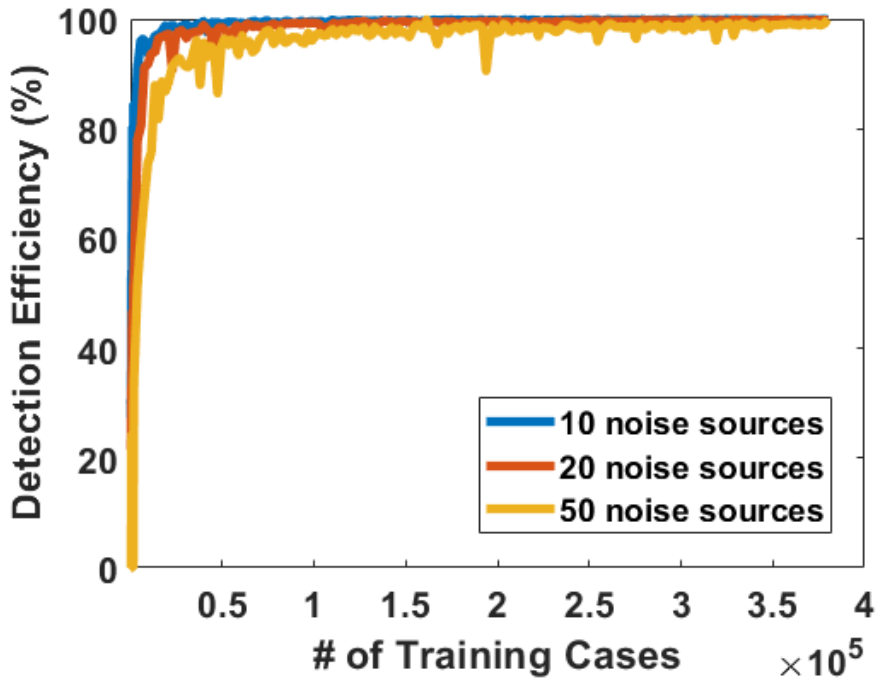

Figure 3: Performance of classifier with additional sources

This experiment is analogous to testing the strength of encryption in the cryptographic community, where the length of the key (number of sources) increases the strength of the encryption and the burden on the attacker, but with enough repetitions, can eventually be deciphered. It is thus concluded that in addition to randomness and statistical properties, the seed of the random number generator must be constantly changed akin to a one-time-pad. The one-time-pad, also known as the Vernam cipher [63], guarantees theoretically perfect secrecy as long as a key is not reused and is randomly generated as proved by Shannon using information theory.

The third and final experiment demonstrates the requirement for an active defense technique to remain covert and not have an impact on the physical process. While other methods such as dynamic watermarking increase the cost of the controller and may thus be detected over multiple runs by simple classification algorithms, covertness may be achieved by constraining the embedded noise using the cost function. A linear time-invariant (LTI) system with stochastic noise is simulated to show that the addition of noise orthogonal to the gradient of the cost function renders noise-based active defenses covert. Over multiple runs, as seen in Figure 4, it is observed that the change in cost when the noise is inserted orthogonally is within the variation in cost due to noise itself, whereas in the case of simple noise addition, the cost is significantly different and can be detected using unsupervised methods such as k-means clustering.

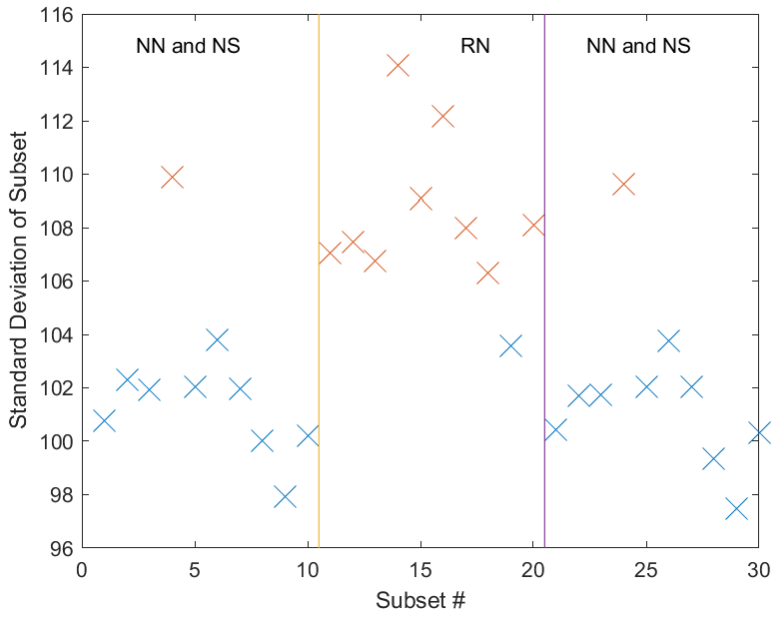

Figure 4: Clustering of controller cost with no noise (NN), random noise (RN), and constrained along gradient (NS)

The results from the three numerical experiments are summarized as follows. Noise-based active defense techniques are a promising counter to pattern discovery and reverse-engineering by AI/ML tools due to their inherent randomness. These techniques may be rendered covert and immune to discovery by AI/ML tools using one-time-pad algorithms that constantly change the source of randomness, perturbing the process variables in a manner that does not impact the process, and preserving the statistical properties of the process variables and their noise before and after the perturbation. These observations lay the foundation for the Covert Cognizance (C2) physical process defense paradigm [64] described in the following chapter. Specifically, zero-impact and zero-observability criteria are developed based on these observations to ensure that the C2 paradigm remains covert and undetectable by even highly knowledgeable adversaries with privileged access such as insiders and APT actors.

2. COVERT COGNIZANCE (C2)

This chapter borrows content from work previously published in the journals Nuclear Technology and Nuclear Science and Engineering [64], [65]. Covert Cognizance (C2) is a novel physical process defense developed for industrial systems that actively perturbs process variables to induce self-awareness (cognizance) and link various subsystems in a discreet manner (covert). The proposed work is a paradigm shift in the philosophy of predictive modeling and levels the playing field between defenders and attackers of critical systems. Existing philosophies have generally been reliant on the ability of the programmer to track code changes, detect intrusion, prevent unauthorized access etc. via specific instructions to the model. The C2 paradigm, on the other hand, seeks to embed awareness in the system itself without needing a human in the loop to make these decisions. Achieving such human-emulating intelligence/awareness, at a minimum, requires the system to have the ability to store, recall, and process experiences at will. In the C2 paradigm, evidence-based records based on the system's execution/operational history are created and embedded along the noise of the process variables to induce awareness across various subsystems. Furthermore, the awareness induced is covert and the evidence-based records are resistant to AI/ML pattern discovery tools since they lie in the noisy space (described as non-observable space in the original manuscript) and are obfuscated to resemble random noise through the use of a one-time-pad algorithm. Since the records are based on the system's own history, they are incorruptible and additionally provide an avenue for intrusion detection and data recovery if the system is compromised.

In this chapter, section 2.1 provides a background of existing predictive modeling philosophies and their response to cyberthreats followed by the motivation behind developing the C2 paradigm in section 2.2. Since the present work focuses on industrial applications of C2, a mathematical framework is developed in section 2.3 to induce cross-cognizance among subsystems in an industrial control system. Lastly, section 2.4 outlines several applications for the C2 work, which is explored in further detail in the following chapters.

2.1 Background

With increasing digitalization of the modern world and the adoption of predictive modeling technology, the fourth industrial revolution, called Industrial 4.0, has integrated the cyber and physical world in industry through the internet of things (IoT) [66], [67]. The Smart Grid , one of the most significant applications of IoT technology, is an advanced digital power system with advanced functionalities such as smart meters, load control, fault detection, self-healing etc. In direct contrast to traditional isolated industrial systems, systems connected to the Smart Grid have vastly benefited from the increased connectivity coupled with advances in data analytics to improve their efficiency and robustness. Nevertheless, the increased connectivity comes at a cost – the increased propensity of cyberthreats. While traditional systems were often air-gapped and isolated from the network, the improved efficiency and optimal use of resources due to the increased connectivity is quite lucrative to businesses and consumers alike, prompting the latter to link their physical infrastructure to the internet through cloud-based services. However, due to the novelty of such technology, lack of best practices, minimal employee awareness, and a host of other human and non-human factors, breaches continue to occur with a sharp uptick in the number of cyberattacks in the past decade. In addition to the leakage of compromising information, financial loss, and identity theft, industrial systems often face the risk of physical damage to their devices as demonstrated by the Stuxnet virus [68] causing irreparable damage to the centrifuges in the Natanz nuclear facility in Iran. Other examples of such instances include the Havex malware targeting U.S. and European industrial systems, BlackEnergy responsible for the 2015 Ukrainian grid cyberattack, Crashoverride/Industroyer responsible for the 2016 cyberattack on the Ukrainian grid, and more recently, the ransomware attack on the Colonial gas pipeline in the U.S. A detailed analysis of similar cyber-incidents on critical infrastructure may be found in the cited NIST report [69]. Such cyberattacks are often highly sophisticated in their implementation; measures are taken to bypass information technology (IT) and operational technology (OT) defenses, obtain privileged credentials through phishing emails, delete files and data pertaining to the grid, and gain access to the supervisory control and data acquisition (SCADA) system to remotely turn off the subsystems. It is evident that there is a need to protect the system at the process level in addition to the IT and OT measures outlined below.

SCADA systems are often highly distributed in the sense they connect various decentralized systems such as water distribution plants, gas pipelines, etc. They were designed

with ease-of-use and robustness in mind with very little regard to security before the advent of IoT and cloud-based services. Legacy systems were typically air-gapped with physical security measures in place and specialized communication protocols to deter most threats. However, with attacks such as the Maroochy-Shire sewage-spill [70] and Stuxnet exploiting SCADA software, the past decade has seen a surge in both IT and OT defenses to secure these systems.

The Purdue Enterprise Reference Architecture (PERA) is an ANSI/ISA-95 industrial standard reference model for the implementation of industrial system architectures [71]–[73]. Through clearly defined hierarchies, it seeks to synchronize business strategy and operational execution through a combination of control, intelligence, and process management to maximize business revenue while minimizing security risks to the system. As illustrated in Figure 5 [74], there are six functional levels to the Purdue model starting from the physical equipment itself to the broader set of internet/web-based services. The innermost level, Level 0, concerns the physical industrial control system comprising of sensors, actuators, pumps etc. that performs the task. Typically, this is the level with the most access to critical processes and the least intrinsic security. Level 1 comprises of devices such as programmable logic controllers (PLCs), relays, microcontrollers etc. that perform control actions via actuators based on sensor measurements. Under the Purdue model, this is the only level that is allowed to directly interact with level 0 equipment albeit with appropriate security protocols. SCADA software and human-machine interfaces (HMI) form level 2 of the hierarchy. These systems utilize the data generated from level 1 to create records (historians), perform data analytics, and issue commands to the PLCs to guide the process. Level 3 comprises of systems whose scope includes the entire industrial plant. These systems form the last layer of OT devices, and their goal is to process lower-level data for the entire site before pushing it upwards for business analytics. The next layer, level 4, comprises local networks for on-site employees, database management systems, internal servers etc. It is prudent that access to the internet does not extend beyond this layer to prevent the possibility of cyberthreats affecting the sensitive physical equipment directly, typically achieved through air-gapping. Finally, layer 5, introduced with the advent of the internet, are typically consumer-facing in that they involve direct interaction with end users. These servers may also host email, backup storage, HR systems etc. IT defenses are typically concerned with securing digital assets in Levels 4 and 5, while OT defenses seek to protect physical assets in Levels 0 – 3 as discussed below.

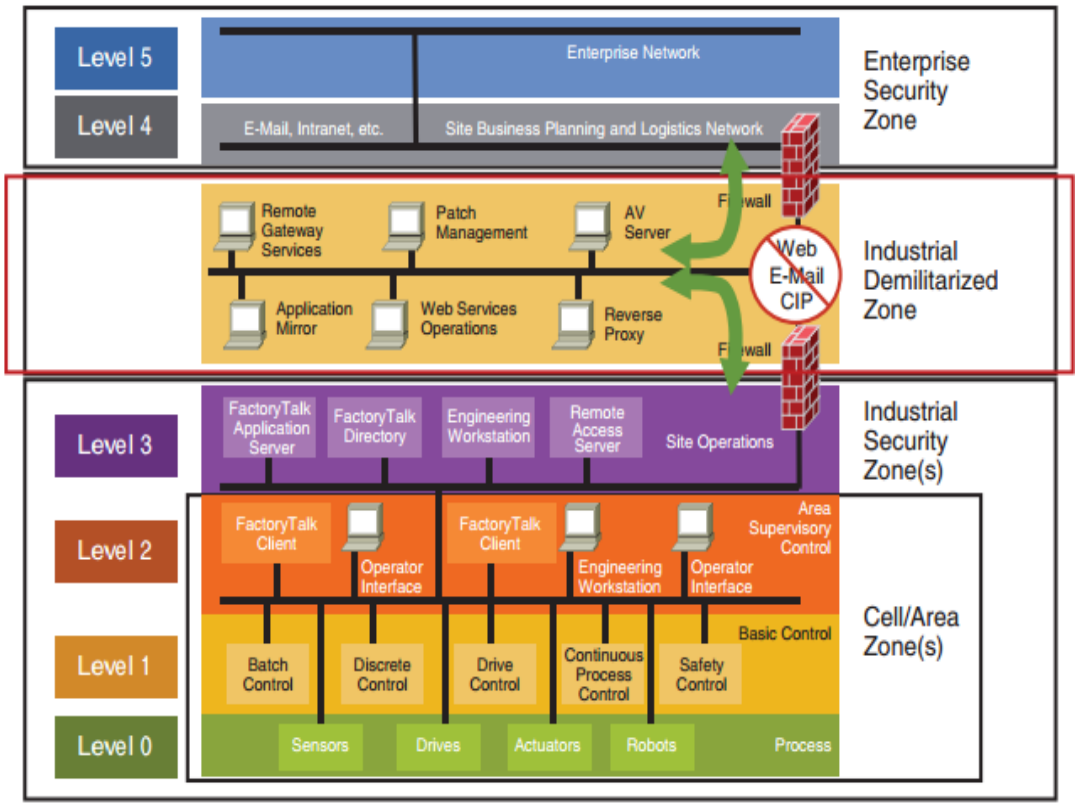

Figure 5: Purdue Enterprise Reference Architecture

IT defenses often involve access-based restrictions such as passwords, firewalls, encryption of network traffic, preventing Internet access etc. The recently emergent adaptive networks [75] automatically configure, monitor, and maintain themselves without the need of humans in the loop. They consist of programmable infrastructure that can continuously adapt to the traffic in the network, switching topologies as necessary through packet switching architecture. The infrastructure utilizes modern data analytics and AI/ML tools to make informed decisions by routing information from users, network elements, instrumentation etc. to a software layer that analyzes the data. Cloud-based technologies such as edge computing [76] provide topological solutions by bringing the data storage and computational facilities closer to the source of the data. While this allows end-users to take ownership of their data, the responsibility of security is also passed onto them. Nevertheless, the topology provides significant advantages with regards to network latency, and proponents argue that the distributed nature of edge computing is more secure since the effects of a single disruption on a network are largely mitigated. Additionally, by keeping the data closer to the source, there is minimal transmission of the data to the cloud thus reducing

the risk of data leaks if the cloud is breached. On the flip side, security requirements near the data source are much higher and sometimes beyond the capabilities of such devices due to resource-based constraints. Note that in the Purdue model, sufficient separation between the layers must be achieved through firewalls and other defenses, and information is not allowed to skip levels, i.e., any communication must pass through the hierarchy in a sequential manner to minimize risks. However, this is often violated in modern systems with IoT devices and cloud-based technologies that often directly link the OT systems to the cloud/internet. If the cloud is breached, the OT systems that often lack intrinsic security features are exposed.

Encryption [77] is another popular IT measure that secures raw data using computationally expensive mathematical algorithms that obfuscate the data using "keys". Similar to the concept of passwords, a key is necessary to reverse the encryption process (decryption) and obtain the raw data, thus preventing unauthorized users from accessing data. The field of cryptanalysis aims to exploit the pseudo-random nature of the mathematical algorithm and other weaknesses in the encryption scheme to decipher the raw data while other methods such as social engineering seek to retrieve the private key information through deceit and clever manipulation of authorized users. In the same vein, blockchain-based technologies have risen over the past decade to meet the demand for secure record-keeping. Popularized by cryptocurrencies such as Bitcoin and Ethereum, the blockchain may be visualized as an append-only ledger where records are hashed with timestamp information before linking it to the previous record. The blockchain is practically immutable since changing a given block requires all subsequent blocks to be modified, making the process computationally expensive as more records are added. The commonly used proof-of-work algorithm to append a record onto a blockchain involves guessing a random number, called a nonce, which when combined with the record creates a hash below a certain threshold value that may be raised or lowered depending on the number of available "miners" in the system. These miners are typically supercomputers that make millions of guesses each second with the first miner to make a successful guess granted the right to append a given record and given a reward as incentive. The result is a fully decentralized and open ledger with easily verifiable records since it is often trivial to verify if the given hash is below the threshold value.

Such solutions often suffer from scalability issues due to the extremely high computational cost of mining [78]. While alternative less-intensive methods have been proposed, these are often less decentralized, require trust-based algorithms, or may be susceptible to low-cost attacks.

Overall, blockchain applications for industrial applications is still in its infancy with some high-profile applications such as tracing diamond sales to prevent counterfeiting, decentralized energy management allowing users to directly buy electricity, secure DNA databases in the medical industry [79]–[81], etc. However, due to the aforementioned concerns of scalability, it is unclear if blockchain technology may prove to be effective for industrial control systems that process massive amounts of data at high frequencies. Additionally, the implementation must also address significant challenges such falsification of the process variables prior to hashing and potential 51% attacks where a majority of the sensors are compromised, as expected from highly skilled adversarial attacks. Other complementary IT measures include biometrics such as fingerprint scanners, iris detectors etc. that restrict access to only authorized personnel based on certain immutable physical features. A well-known shortcoming of such methods, however, is the risk of insider threat where adversaries typically have the necessary privileges to infiltrate the system and bypass biometrics-based safety measures, necessitating a security paradigm to protect the system at the process level known as OT defenses.

. The primary objective of OT measures is to always ensure the safe operation of all physical assets even if the system is compromised. In contrast to IT measures that treat the physical assets as a black box and focus on access prevention, OT defenses protect these assets by leveraging the physics of the system to create model-based defenses and detect discrepancies in the process variables. Digital twin [82] technology has become increasingly popular to create a virtual version of the physical system that serves as a reference for anomaly detection, predictive maintenance etc. The key to such measures is the fidelity and proprietary nature of the digital model, i.e., it is assumed that the attackers do not have access to the digital twin or a similar model. However, this assumption has been proven to be naïve in recent years with the advent of AI/ML techniques as outlined in Section 1.2 that are able to learn the model with an initial approximation [39]. Additionally, insider attacks often come from within the system, dispelling any notions of access-based security.

A growing body of research has been specifically devoted to the class of attacks known as false data injection attacks (FDIAs) [61], [83]–[85]. FDIAs seek to send a target industrial system along a different, and often dangerous, trajectory while appearing to be under normal operation to an observer through clever falsification of the process variables. In the seminal paper on FDIAs against Smart Grids [86], the authors exploit the dynamics of the system to ensure that the attack

remains covert and undetectable through typical statistical methods such as χ^2 residual-based detectors. It is assumed that perfect knowledge of the grid configurations is available to the attacker, which is typical of insiders and APT actors. Even under imperfect conditions, it has been demonstrated that FDIAs may be constructed to bypass χ^2-detectors [87]. The goal of FDIAs may be multifaceted – they may be employed to cause physical damage to equipment [88], defraud bulk electricity markets by manipulating locational marginal prices, profit off demand response systems [89] etc. Demand response systems are often an attractive target since they often have many participants that make it difficult to attribute cyberattacks. Using representative models, optimal attack strategies have been explored to generate sudden overload spikes in the data to cause physical damage to the equipment while subtly avoiding detection or manipulate the market through clever integrity attacks on the pricing signals.

To combat these attacks, several methods have been proposed depending on the model, attacker knowledge, operational conditions etc. Security analysis of demand response systems indicate that dynamic pricing methods are more resilient to such adversarial attacks than direct load control. Generalizing to industrial control systems, under noiseless conditions, decoders have been designed to render the system resilient to a limited number of FDIAs [90]. However, the computational complexity of the decoder algorithm and the presence of real-world noise are major drawbacks to real-life implementation. Other statistical methods include discrepancy-based methods that exploit the difference in distribution of clean and falsified data through the Kullbeck-Liebler divergence metric [91]. However, the effectiveness of such methods is severely degraded in the presence of various sources of noise in a real system since the attacks may be subtler than the deviations due to noise.

Few others in the control community have taken a mitigation-based approach to limit the impact of FDIAs on industrial systems [50]. For example, PCA-based methods have been proposed to detect cyberattacks that disrupt correlations between the process variables [31], [36]. This limits a successful cyberattack to operating within the uncorrelated space to avoid detection. However, it is recognized that if the adversary has access to all sensors and actuators, it is possible to manipulate the measurements in a manner that preserves correlations and evade model-based schemes. The replay attack, popularized by the Stuxnet virus, exploits this observation by replaying previous steady-state data while sending the system along a different trajectory. Since past data is representative of the system and preserves all the necessary correlations, it appears

genuine to most model-based algorithms, necessitating the use of active techniques that interrogate the system through perturbations that may be traced for authenticity. A popular example of active techniques is dynamic watermarking [61]where a random signal based on a hidden Markov model is added to the actuator input that may be traced through the output of the signal using a modified χ^2-detector. The efficacy of such active techniques was previously discussed in Section 1.4 in further detail.

The ISA/IEC 62443 series by the Global Cybersecurity Alliance provides a metric called the security level to evaluate the security of industrial systems [92]. This was subsequently adapted in the ISA-99.01.01 standard as security assurance levels. The metric is based on seven foundational requirements, namely, access control, use control, data integrity, data confidentiality, restrict data flow, timely response to an event, and resource availability. In this model, systems are assigned security levels in terms of the protection they offer against adversaries with varying resources, skills, means and motivation. Systems with level 1 security offer protection against casual or coincidental violations that are typically unintentional and expected to occur during routine operation. A few examples of such violations include an operator accessing the wrong PLC, accidentally setting an incorrect setpoint outside the permissible limits etc. Protection against intentional violations that are simple to cause and require low motivation and skill satisfies the requirements for level 2. For example, an adversary may use a publicly known exploit to gain access into the OT systems, or simply send a virus to an email server that spreads to various systems due to unsuspecting employees, requiring very low effort on the side of the adversary. Systems that have level 3 protection are difficult to penetrate and successful attacks require significant skill and resources to execute. These adversaries are typically motivated, possess system-specific knowledge and may even penetrate the IT/OT barrier to cause damage to physical assets. Lastly, level 4 requires systems to be protected against adversaries that are highly resourceful and motivated such as state-sponsored attackers, APT actors, and insiders. This represents the highest degree of sophistication for a cyberattack and requires intimate knowledge of the target system, its vulnerabilities, and the ability to develop zero-day exploits like the Stuxnet virus.

Summarizing from the literature review on existing methods, it is evident that there is insufficient protection at the process level against sophisticated threats. In other words, there is a need to develop a physical process defense paradigm that can confer level 4 protection onto

industrial systems such that the system is rendered resilient to adversaries that possess intimate knowledge of the system, motivating the development of the C2 paradigm in the following section. C2 is intended as a complementary layer of defense to existing IT and OT protocols to protect data integrity at the process level.

2.2 Motivation

The chief motivation behind the development of the C2 paradigm is rooted in the observation that existing IT and OT methods are inadequate in addressing sophisticated adversaries such as insiders and APT actors to protect the data at the process level, i.e., Level 0 of the Purdue model. The playing field is often lopsided in favor of the attackers as demonstrated by the simplicity of the replay attack and recent research suggesting that data falsification attack payloads may be delivered with as little as 2 KB of memory using simple runs analysis and line segments to learn the behavior of the process variables[93]. In addition to being bypassed by adversaries with privileged access, existing defenses such as model-based passive defenses and noise-based active defenses are often probabilistic in nature with undesirable false positive rates. The C2 physical process defense fills this gap by providing a deterministic solution to the problem of FDIAs and insider threats through the evidence-based records embedded along the process variables. Effectively, the system is alert to how it is typically operated, and thus any falsification of the data is detected, and the location of the intrusion is pinpointed.

C2 is a paradigm shift in the approach towards cybersecurity of industrial systems that seeks to level the playing field or even tilt it in the favor of defenders. As the name implies, it seeks to embed intelligence in industrial systems in a covert manner that avoids some of the common pitfalls associated with model-based defenses and active techniques. It operates under the assumption the attacker has full privileged access to a given industrial system and may manipulate its process variables in any manner. The heart of the C2 physical process defense lies in the somewhat paradoxical observation that one may perform an FDIA on the system with the opposite goal, i.e., instead of sending the system along a different trajectory to damage the system, the process variables may be falsified with the goal of securing the system instead. In other words, the process variables are falsified to prevent further falsification.

The key to the C2 paradigm lies in the inherent redundancy of most industrial systems. Physical models and digital twins often have thousands of variables coupled to each other through

physics-based equations that constrain them to only a few active degrees of freedom (DOFs) describing the dominant behavior of the process. For instance, in a nuclear power plant, the temperatures of the reactor at various points in the core, inner, and outer plenum are highly correlated across time and their temporal evolution may be reduced to a handful of dominant directions. The other uncorrelated and independent directions are described by variations due to noise in the sensors and other disturbances that do not affect the temporal evolution of the system in a meaningful manner. This leaves the designer of a predictive model with a vast space spanned by the numerous non-influential DOFs that may serve as courier variables to embed the evidence-based records. It is highlighted here that this inherent redundancy is what enables the C2 paradigm to remain covert; it only uses existing process variables without requiring the use of additional log files, variables etc. that may leave a footprint on the system and does not affect the system behavior through the embedding. The above discussion may be summarized through two key constraints, namely, zero-impact and zero-observability, as described in further detail in the later sections.

While the C2 paradigm has many applications, the focus of the present work is on constructing an additional layer of physical process defense to protect industrial systems at the process data level from insiders and adversaries possessing domain knowledge such as APT actors. As such, a framework is developed in the context of industrial systems in the following section.

2.3 C2 Framework for Industrial Systems

The C2 physical process defense is founded on the principles of zero-impact and zero-observability. In the context of industrial systems, zero-impact implies that the induced awareness must not leave a footprint on the system or affect the process variables in any significant manner that causes the system to deviate from normal operation. Note that this is markedly different from active techniques such as dynamic watermarking that sacrifice optimality of the control system for improved detection and have associated false positive rates. Zero-observability, as the name implies, requires the C2 implementation to be covert and undetectable by a third party. In essence, a system with and without the C2-enabled awareness must operate identically in identical conditions. It must be resistant to pattern discovery by AI/ML algorithms as validated in the following chapter. Due to the inherent redundancy of industrial system, this is achieved by using the dominant directions of the process variables of one subsystem to create evidence-based records and embedding the latter along the noise of the process variables of another subsystem to render

the two subsystems cognizant of each other – a process known as cross-cognizance. If the statistical properties of the noise are preserved, the embedding does not impact the operation of either subsystem while achieving cognizance goals.

The first step to creating evidence-based records is to extract the dominant and noisy directions from the process variables, also called the observable and non-observable space respectively. The decomposition of industrial data into the two subspaces may be achieved through reduced-order modeling (ROM) techniques such as principal component analysis (PCA), proper orthogonal decomposition, randomized linear algebra techniques etc. The dominant DOFs, or active DOFs, span the observable space, and carry most of the information on the evolution of the process variables with time and their correlations with each other. In fact, model-based passive defense techniques often rely on these DOFs to construct an approximate physical model and test for statistical deviations. The complementary subspace, i.e., the non-observable space, is spanned by non-dominant DOFs that often carry information on the unexplained variance in the data, typically due to noise in the system. These DOFs serve as courier variables to carry the information in the evidence-based records. Mathematically, consider the model of an industrial system with state variables $\mathbf{x}$, control input $\mathbf{u}$, and output $\mathbf{y}$ whose temporal evolution with time t is described by functions $f(\cdot)$ and $g(\cdot)$ as stated in Eqs. 2.1-2.4. Let the observable and non-observable space of the state extracted using ROM techniques be given by the orthonormal matrices $\mathbf{Q}$ and $\mathbf{Q}^{\perp}$ respectively where $\perp$ denotes orthogonality, i.e., $\mathbf{Q}\mathbf{Q}^{\perp} = \mathbf{0}$. Additionally, consider two subsystems A and B, denoted by variable subscripts, that must be linked with each other through the C2 implementation.

$$\dot{\mathbf{x}}_{\mathrm{A}} = f_{\mathrm{A}}(\mathbf{x}_{\mathrm{A}}, \mathbf{u}_{\mathrm{A}}, t) \tag{2.1}$$

$$\mathbf{y}_{\mathrm{A}} = g_{\mathrm{A}}(\mathbf{x}_{\mathrm{A}}, \mathbf{u}_{\mathrm{A}}, t) \tag{2.2}$$

$$\dot{\mathbf{x}}_{\mathrm{B}} = f_{\mathrm{B}}(\mathbf{x}_{\mathrm{B}}, \mathbf{u}_{\mathrm{B}}, t) \tag{2.3}$$

$$\mathbf{y}_{\mathrm{B}} = g_{\mathrm{B}}(\mathbf{x}_{\mathrm{B}}, \mathbf{u}_{\mathrm{B}}, t) \tag{2.4}$$

Then, the evidence-based records are extracted from the output $\mathbf{y}$ of one subsystem and obfuscated using a one-time-pad algorithm to resemble statistically random noise, represented using the composite function $h(\cdot)$ here for simplicity. These records are then stored along the process variables of the other subsystem through perturbations $\Delta\mathbf{x}$ in accordance with the C2

paradigm. Note that the embedding is done from subsystem A to B and vice-versa as shown in Eqs. 2.5 and 2.6 to induce cross-cognizance among the subsystems. The zero-impact and zero-observability constraints are satisfied via Eqs. 2.7-2.10 to ensure that the perturbations do not affect the system behavior and cannot be detected by AI/ML pattern discovery tools. Specifically, Eqs. 2.7 and 2.8 dictate that the perturbation does not affect the dominant behavior of the system, i.e., its effects do not permeate into the observable space of the process variables to ensure covertness. Eqs. 2.9 and 2.10 describe the zero-impact condition if one wishes to eliminate the effect of the perturbation on the output of the control system.

$$\mathbf{Q}_{\mathrm{A}}^{\perp}(\mathbf{x}_{\mathrm{A}} + \Delta\mathbf{x}_{\mathrm{A}}) = h_{\mathrm{A}}(\mathbf{y}_{\mathrm{B}}) \tag{2.5}$$

$$\mathbf{Q}_{\mathrm{B}}^{\perp}(\mathbf{x}_{\mathrm{B}} + \Delta\mathbf{x}_{\mathrm{B}}) = h_{\mathrm{B}}(\mathbf{y}_{\mathrm{A}}) \tag{2.6}$$

$$\mathbf{Q}_{\mathrm{A}}\Delta\mathbf{x}_{\mathrm{A}} = 0 \tag{2.7}$$

$$\mathbf{Q}_{\mathrm{B}}\Delta\mathbf{x}_{\mathrm{B}} = 0 \tag{2.8}$$

$$g_{\mathrm{A}}(\mathbf{x}_{\mathrm{A}} + \Delta\mathbf{x}_{\mathrm{A}}, \mathbf{u}_{\mathrm{A}}, t) = g_{\mathrm{A}}(\mathbf{x}_{\mathrm{A}}, \mathbf{u}_{\mathrm{A}}, t) \tag{2.9}$$

$$g_{\mathrm{B}}(\mathbf{x}_{\mathrm{B}} + \Delta\mathbf{x}_{\mathrm{B}}, \mathbf{u}_{\mathrm{B}}, t) = g_{\mathrm{A}}(\mathbf{x}_{\mathrm{B}}, \mathbf{u}_{\mathrm{B}}, t) \tag{2.10}$$

Note that the use of a one-time-pad algorithm is critical in this implementation. This is because for steady-state, periodic or saturated processes, the same temporal information may be repeated across various time-intervals, resulting in the same output if the one-time-pad is reused. As demonstrated in section 1.4, this may be detected with sufficient samples by an insider familiar with the implementation of the C2 paradigm especially if the embedding occurs at every time-step for high frequency data. Keeping in spirit with the C2 paradigm, the source of the one-time-pad may be found within the system itself without requiring the use of additional variables or log files. In fact, physical devices often gain sufficient entropy during operation and are excellent examples of true random number generators that are cryptographically secure. Common examples include /dev/urandom on UNIX-based systems and hardware random number generators based on thermal noise.

Interestingly, it is observed that Eqs. 2.9 and 2.10 demonstrate the potential for a non-observable space in the function $g(\cdot)$ that may be exploited for further obfuscation of the C2 process. Similarly, one may extend this to the state function $f(\cdot)$ and the controller cost J to ensure no impact on any hidden state variables. The above setup is only one instantiation of the C2

paradigm and further constraints on other process variables may be designed depending on the target industrial system. Although not explored in this work, it is possible to construct a non-observable space that encompasses all possible variations and dependencies in the system using active subspace identification techniques from ROM. In fact, this highlights the flexibility that the C2 paradigm offers to a defender due to the vast number of non-influential/noisy DOFs resulting from the inherent redundancies of the system.

An important question that may arise during the discussion of insider threats is whether the designer of the C2 system may bypass it. This may be addressed through the use of random embedding to render it impossible to predict when the C2 physical process defense is active. This relies on the observation that since FDIAs are subtle and occur over long timeframes to bypass existing defenses, it is not necessary to embed evidence-based records at every time-step. In fact, the records may be generated from previous time-steps and embedded in the process variables at a later time-step while continuously being obfuscated by a one-time-pad algorithm to prevent reverse-engineering. The whole process may be randomized through independent and random pulses that dictate when the record generation and embedding should take place, and the corresponding process and courier variables. Further architectural considerations may be employed to render the implementation covert; however, this is beyond the scope of this manuscript which focuses on software-based implementations.

2.4 Applications of C2

The C2 framework for industrial systems has several applications in anomaly detection, data recovery, data deception through decoys, sensor fingerprinting etc. This section provides a brief overview of how the embedded evidence-based records may be utilized to achieve these predictive modeling functionalities while remaining covert to humans and AI/ML-based pattern discovery tools. The incorruptible records ensure that the system is aware of its own execution history and can adapt to changes in operation. In the area of anomaly detection, system anomalies may arise due to natural factors (e.g. external disturbances, equipment malfunction) or malicious intrusion. Here, C2 may be used to differentiate between naturally occurring anomalies and malicious intrusion since the latter involves an element of falsification that can be detect using the embedded evidence-based records. If the C2 paradigm is implemented in a distributed industrial system with multiple sensors, the evidence-based records from critical sensors may be embedded

throughout the system due to the vast number of non-influential DOFs available to serve as courier variables. In this case, even if the critical sensors are compromised, the C2 embedding provides an avenue for data recovery that does not rely on vault-like data storage mechanisms whose security is predicated on the lack of access – a naïve assumption considering insider threats.

The C2 paradigm may also be used to embed constantly changing signatures/watermarks in the data for fingerprinting akin to steganography. Contrasting with existing fingerprinting techniques, the C2 fingerprint is resistant to pattern discovery due to the use of a one-time-pad and the validation cannot be bypassed by statistically similar data. Regarding reverse-engineering applications, the idea of observable and non-observable space may be extended to decompose industrial data into two sets of metadata describing the underlying physical model and the process parameters respectively. The metadata may be obfuscated and manipulated for data masking purposes to enable the sharing of sensitive data while minimizing the risk of exposure in the event of a leak. Additionally, benchmark datasets may be constructed for AI/ML applications by preserving the inferential properties of the data and fusing it with different physical systems. Lastly, the C2 paradigm may be used in an offensive setting. Using the example of a drone, it may be beneficial to intentionally induce patterns in the embedding to mislead attackers into a false sense of victory when discovered. These applications are discussed in further detail below and subsequent chapters of the manuscript demonstrate some of these capabilities in representative industrial systems.

2.4.1 Intrusion Detection

Generally, anomalies in industrial systems may be classified as natural anomalies like sensor drift that arise due to external disturbances, equipment degradation etc. or man-made like FDIAs that arise due to malicious intrusion [31]. Such anomalies typically manifest themselves as sudden or subtle deviations in the data depending on the timescale on which they occur. Apart from clever FDIAs that exploit the non-observable space, these anomalies may be detected by passive model-based methods using the so-called lower-order and higher-order components (LOCs and HOCs) that carry information about the correlations among process variables. Drawing parallels with the C2 terminology, the LOCs and HOCs lie in the observable space of the process variables and carry information about their temporal evolution. Extracted using SVD-based algorithms, LOCs are sensitive the dominant behavior while HOCs are sensitive to subtle

variations that occur over longer time-intervals. Unlike HOCs, LOCs are generally robust to variations due to noise, and as such are an ideal candidate for AI/ML features. On the other hand, HOCs are easily obfuscated by noise and require careful denoising to extract the critical information relevant for AI/ML-based detection algorithms. Consequently, denoising algorithms have been developed that place an emphasis on preserving LOCs and HOCs for AI/ML applications to improve classification performance of such models in the presence of noise. Preliminary analysis of classifier performance on reactor datasets indicates that such algorithms may outperform other filtering algorithms such as moving average, exponential smoothing, Fourier smoothing etc. with regards to classification ability while also providing good estimates of the process variables in the mean-squared sense. Furthermore, they can continually update estimates as more measurements are gathered and may be deployed for real-time condition monitoring and anomaly detection applications.

In addition to detection, there arises a need to classify the anomaly as natural or man-made since the response required is radically different in the two cases. Anomalies arising out of equipment degradation often require replacing/repairing of physical components while malicious intrusion requires response from a cyberteam to analyze possible threat vectors, attacker intent etc. and to mitigate the impact of the cyberattack. Here, it is noted that the key difference between the two types of threats is that the process variables are often falsified in cyberattacks while they respect the system dynamics in the case of natural disturbances. Thus, after a preliminary anomaly detection through passive model-based methods, the C2 physical process defense layer may be used to distinguish between the two and provide an appropriate response. Even if the cyberattack is statistically subtle enough to bypass model-based defenses, it is still detected by the C2 defense since the latter does not rely on statistical methods and instead provides a deterministic solution through the embedding of evidence-based records. This is implemented in chapter 3 in a representative linearized nuclear reactor modeled as an LTI system and validated using both statistical and AI/ML methods.

2.4.2 Data Recovery

A common critique of existing IT and OT defenses is that once the system is compromised, defenders are typically relegated to shutting down the system and relying on backups to restore operation, resulting in massive downtime as evidenced by the Colonial pipeline ransomware attacks [94]. Additionally, there is no avenue for data recovery in such mechanisms and backup data is often stored in vault-like systems or implemented through redundant sensors whose success hinges upon the assumption that they cannot be accessed/falsified by the attacker. However, with the increasing propensity of cyberattacks, it may not necessarily be practical to constantly shut down the system, thus establishing the need for a more robust mechanism that allows the system to continue operation even while compromised. This idea is captured through the concept of energy resiliency in power grid infrastructure, where the goal is to continue delivering energy to critical loads and mitigate the impact of outages via self-recovery mechanisms during unforeseen events. The C2 paradigm fulfills this critical need by embedding information about the system's own operational history in the process variables themselves via evidence-based records.

While the decentralized/distributed nature of industrial systems is often cited as a drawback for security, C2 turns this into an advantage by inducing cross-cognizance between loosely coupled systems. For instance, information about a critical component such as the core of the nuclear reactor may be embedded in the process variables of an isolated pump, providing an avenue for data recovery if the process variables in the core are falsified. Furthermore, due to the vast number of non-influential DOFs available as courier variables, multiple copies of the same embedding information may be stored with appropriate obfuscation using different one-time pads to ensure that no correlations are accidentally induced violating the covertness requirement. Even if multiple sensors are compromised, it is possible to recover the lost information from the distributed system to continue operation while a cyberteam responds to the intrusion and falsification. This is especially important to remotely controlled and automated industrial systems that must remain functional but may not be easily accessible in times of distress. As an example, a representative small modular reactor (SMR) is simulated using the modeling software Dymola with C2 modules developed for automated intrusion detection and data recovery in chapter 4.

2.4.3 Process Fingerprinting

Deviating from control systems, the C2 paradigm also has steganographic applications that may be leveraged for covert watermarking of other processes. For instance, the mathematical framework developed in section 2.3 is also applicable to computational solver algorithms such as Newton's method or Krylov subspace methods. Developers for commercial solver software may use the C2 embedding methodology to fingerprint their solvers by generating evidence-based records from the solver execution history (date, time, initial conditions etc.) and embed these records along the solution itself (self-cognizance) to serve as an authentication of the solver execution. Additional constraints may be imposed on the embedding to ensure that the perturbations carrying the fingerprint lie in the non-observable space of the solver algorithm and do not have any impact on the final solution. For instance, if a commercial solver uses Newton's method in one of its sub-calculations, a proprietary watermark such as the logo or the name of the solver may be obfuscated using a one-time-pad derived from the system's inherent randomness (thermal noise, system time etc.) to create evidence-based records. Using active subspace methods, one may construct a subspace spanned by the gradient of the solution with respect to minor perturbations and embed along the subspace orthogonal to the gradient to ensure minimal impact on the solution of the solver, akin to the idea using the cost function in industrial control systems.

The application may also be extended to fingerprinting sensors through the concept of self-cognizance, i.e., the evidence-based records are generated from and embedded in the same sensor. This may be done through ROM techniques such as dynamic mode decomposition, singular spectrum analysis, randomized window decomposition etc. that exploit the temporal correlations and measurement noise within the individual sensor to construct the non-observable space for embedding. By way of example, a sensor in a representative control system equipped with a Kalman filter is perturbed to carry information about itself in this preliminary work [95]. Here, the residual between the perturbed output of the sensor and the Kalman filter estimate is used as a courier variable to embed the obfuscated records derived from the previous sensor measurements with additional constraints imposed that the perturbated output is covert to statistical χ^2-detectors and does not impact the control input. The process may be validated via residual analysis of the output and the filter estimate to recover the embedded information, thus serving as a fingerprint.

2.4.4 Data Masking

The decomposition of industrial data into the observable and non-observable space using active subspace identification ROM techniques for C2 motivates an interesting discussion on the physical meaning of these subspaces. While the non-observable space is typically taken to represent unexplained variances such as noise, the observable space spanned by the active DOFs carry information on the dominant behavior of the process variables. The observable space may be further decomposed into the so-called fundamental and inference metadata, where the fundamental metadata describes the governing physical principles of the system while the inference metadata carries information about the operational parameters. For instance, the fundamental metadata of a reactor experiencing a power increase may be described using an exponential profile, whereas the corresponding inference metadata may be the time constant, saturation power etc. that represent process parameters. Since datasets typically vary only in their process parameters, AI/ML algorithms may be fed the inference metadata for the purposes of optimization, regression, classification etc.

The following body of work is motivated by two key observations from the decomposition of data into its fundamental and inference metadata. Firstly, it is observed that replacing the fundamental metadata of the dataset with that of a different system while preserving the inference metadata effectively masks the system identity since any association with the origin of the data is captured in its underlying physics, i.e., the fundamental metadata. This process, called the deceptive infusion of data (DIOD) [37], enables the sharing of sensitive data such as that from nuclear reactors with third-party AI/ML services without the risk of data leaks that tie it to the reactor. It may also be used to prevent reverse-engineering of datasets to obtain sensitive information from critical systems as done by tools like SINDy, PDE-FIND etc [51], [52]. Secondly, multiple datasets may be generated by using the same inference metadata but different fundamental metadata from several generic/non-critical systems to create a benchmark for AI/ML algorithms. Based on the concept of mutual information, an ideal AI/ML algorithm is expected to be invariant to the extraneous information carried by the fundamental metadata and must make the same inference on all datasets since the inference metadata is identical in all cases. Mutual information is a symmetric measure capable of quantifying all linear and non-linear dependencies between systems. It represents the upper limit of separability in classification applications and is a useful tool in identifying relevant and irrelevant features for AI/ML tasks. Consequently, several

constraints may be developed to preserve the inference metadata for AI/ML applications and obfuscate the fundamental metadata using mutual information depending on the level of obfuscation required. These applications are explored in further detail in chapter 5.

The following body of work is motivated by two key observations from the decomposition of data into its fundamental and inference metadata. Firstly, it is observed that replacing the fundamental metadata of the dataset with that of a different system while preserving the inference metadata effectively masks the system identity since any association with the origin of the data is captured in its underlying physics, i.e., the fundamental metadata. This process, called the deceptive infusion of data (DIOD) [37], enables the sharing of sensitive data such as that from nuclear reactors with third-party AI/ML services without the risk of data leaks that tie it to the reactor. It may also be used to prevent reverse-engineering of datasets to obtain sensitive information from critical systems as done by tools like SINDy, PDE-FIND etc [51], [52]. Secondly, multiple datasets may be generated by using the same inference metadata but different fundamental metadata from several generic/non-critical systems to create a benchmark for AI/ML algorithms. Based on the concept of mutual information, an ideal AI/ML algorithm is expected to be invariant to the extraneous information carried by the fundamental metadata and must make the same inference on all datasets since the inference metadata is identical in all cases. Mutual information is a symmetric measure capable of quantifying all linear and non-linear dependencies between systems. It represents the upper limit of separability in classification applications and is a useful tool in identifying relevant and irrelevant features for AI/ML tasks. Consequently, several constraints may be developed to preserve the inference metadata for AI/ML applications and obfuscate the fundamental metadata using mutual information depending on the level of obfuscation required. These applications are explored in further detail in chapter 5.

2.4.5 Deception

To quote Almeshekah and Spafford [96], “achieving security cannot be done with single, silver-bullet solutions; instead, good security involves a collection of mechanisms that work together to balance the cost of securing our systems with the possible damage caused by security compromises and drive the success rate of attackers to the lowest possible level.” Deception is a security mechanism popularized in the computer science community and IT defenses that involves the intentional misdirection of adversaries with irrelevant information while protecting critical

information. Entire networks may be constructed to serve as decoys and honeypots that do not impact the real system while providing a ruse for the adversarial interaction. Analogously, the vast number of non-influential DOFs in modern industrial systems present the defender with the option to mislead attackers through intentional embedding of false information along these DOFs. Using the example of the drone introduced at the beginning of chapter 1, misleading information may be directly embedded without obfuscation along some of the non-influential DOFs to misdirect adversaries while protecting critical information using other DOFs.

In essence, this is a honeypot strategy to thwart reverse-engineering of critical systems using independent DOFs that do not affect system behavior while preserving system functionalities using the DOFs embedded with the evidence-based records. Pattern discovery tools are expected to find the misleading information due to the lack of obfuscation, thus lulling the adversaries into a false sense of success. Additional layers may be added through partial obfuscation to render the task difficult but not impossible for AI/ML pattern discovery tools to make the honeypot appear more convincing and further lure the adversary. Distinct from other deception methods, the C2 paradigm does not involve the creation of separate networks, additional log files etc. and embeds both the evidence-based records and the fake information in the process variables themselves. While this manuscript focuses on defensive applications such as intrusion detection and data recovery mentioned above, the proposed angle enables C2 to be used in an offensive setting putting the onus on the adversary to decipher whether the extracted information is legitimate.

3. CASE STUDY: INTRUSION DETECTION

In this chapter, the C2 physical process defense is implemented and validated in a linearized model of a nuclear reactor (chosen as a representative LTI industrial system) in MATLAB using the mathematical framework developed in section 2.3. The implementation is validated visually using histograms, plots of the state variables, response, control inputs, and statistical tests to ensure that the zero-impact constraints are satisfied. Subsequently, AI/ML-based pattern discovery tools are trained on the reactor data to ensure that temporal correlations in the data are preserved and that the embedding does not introduce additional features that violate the zero-observability constraints. The goal of this chapter is to provide a deterministic solution to intrusion detection and differentiate it from other naturally occurring anomalies via the C2 paradigm when existing IT and OT measures have been bypassed.

The chapter is organized as follows: section 3.1 provides the problem setup for a linearized reactor model and the associated C2 constraints. Section 3.2 covers the results of the embedding and demonstrates the ability of the C2 physical process defense to detect intrusion against FDIAs. The implementation is validated statistically in section 3.3 and using a state-of-the-art AI/ML tool in section 3.4.

3.1 Problem Setup

The nuclear reactor consists of two subsystems, namely, the core (A) and the steam generator (B), in which cross-cognizance is induced through the C2 embedding. The goal of the embedding is to provide an additional layer of defense at the process level in case the core or steam generator process variables are falsified. The state variables of the reactor core are the reactor power, precursor power, fuel temperature, and moderator temperature, of which the reactor power is the measured output. The core is augmented with a Kalman filter [97] to provide an optimal estimate of the state and guide the controller to reach a target power output using the inlet temperature as control input and a setpoint value. The core power is chosen as the output variable and is corrupted by additive zero-mean white Gaussian noise $\boldsymbol{\epsilon}_A$ with a known covariance matrix $\mathbf{R}_A$. The given LTI system may be formulated as a linear-quadratic-Gaussian problem with the Kalman filter providing an optimal estimate of the state in a linear sense and the Kalman gain is

set to optimize the control process with respect to an objective cost function J that depends on the state variables and the control input.

With regards to the steam generator, the state variables are the primary temperature, metal temperature, water level, downcomer enthalpy, steam quality, and pressure. Of these, the primary temperature, water level, and the pressure are the measured outputs corrupted by additive zero-mean white Gaussian noise $\boldsymbol{\epsilon}_{\mathrm{B}}$ with a known covariance matrix $\mathbf{R}_{\mathrm{B}}$ and the system evolves according to the input moderator temperature from the core. Although no control algorithm was involved, the steam generator is also augmented with a Kalman filter to provide estimates of the state variables using the output. The equations of the core and steam-generator are provided in state-space form in Eqs 3.1-3.4 with further information in the Appendix. The reactor is simulated multiple times in MATLAB by varying the operating conditions and the seed number used to generate the noise, using which the observable and non-observable spaces are constructed for the two subsystems.

Since the linear-quadratic-Gaussian controller provides a function relationship between the control input and the state variables via the Kalman gain, the controller cost may be written solely as a function of the state variables, and the gradient with respect to the state variables represents the direction of greatest change for a small perturbation. Geometrically, the direction(s) orthogonal to this direction then represents the direction of minimal change, i.e., a level curve where the controller cost is unperturbed. Thus, an additional constraint may be imposed when constructing the non-observable space as seen in Eq 3.5.

$$\mathbf{x}_{\mathrm{A}} = \mathbf{F}_{\mathrm{A}}\mathbf{x}_{\mathrm{A}} + \mathbf{G}_{\mathrm{A}}\mathbf{u}_{\mathrm{A}} \tag{3.1}$$

$$\mathbf{y}_{\mathrm{A}} = \mathbf{H}_{\mathrm{A}}\mathbf{x}_{\mathrm{A}} + \boldsymbol{\epsilon}_{\mathrm{A}} \tag{3.2}$$

$$\mathbf{x}_{\mathrm{B}} = \mathbf{F}_{\mathrm{B}}\mathbf{x}_{\mathrm{B}} + \mathbf{G}_{\mathrm{B}}\mathbf{u}_{\mathrm{B}} \tag{3.3}$$

$$\mathbf{y}_{\mathrm{B}} = \mathbf{H}_{\mathrm{B}}\mathbf{x}_{\mathrm{B}} + \boldsymbol{\epsilon}_{\mathrm{B}} \tag{3.4}$$

$$\frac{\partial J}{\partial \mathbf{x}_{\mathrm{A}}} \cdot \Delta\mathbf{x}_{\mathrm{A}} = 0 \tag{3.5}$$

Due to the limited number of variables, the present model only has two active DOFs each for the core the steam generator, leaving two and four noisy DOFs respectively to serve as courier variables. Consequently, evidence-based records are created from the power of the core and

embedded along the non-observable space of the steam generator state estimates and vice-versa using the water level of the steam generator and the non-observable space of the core state estimates. Note that a real nuclear reactor is expected to have thousands of sensors resulting in a much larger number of noisy DOFs that can carry much more information. Using Eqs. 2.5-2.8, the state estimates of the Kalman filter of both the core and the steam generator are perturbed in accordance with the zero-impact and zero-observability constraints of the C2 paradigm.

The C2 implementation is subsequently validated using statistical measures and AI/ML pattern detection tools to ensure that the zero-impact and zero-observability constraints are satisfied. The following section summarizes the results from this numerical experiment and validates it statistically and against AI-based pattern detection.

3.2 Results

This section describes the results from the C2 implementation in the linearized reactor model. In this numerical simulation, evidence-based records are generated from the core and steam generator and embedded in each other in real-time at discrete intervals of 1 second. The process is simulated over a period of 60 seconds until the reactor reaches the setpoint power of 2100 MW, an increase of 100 MW from the initial equilibrium of 2000 MW around which the core and steam generator equations were linearized. Additionally, a reference case with the same random number generator seed (for noise) is simulated without the C2 implementation to compare and validate the zero-impact and zero-observability criteria.

In the reactor model, the evidence-based records from the water level and core power are initially patterned since they describe the temporal evolution of the two process variables. Using a one-time-pad, the information is obfuscated via permutations and scaling to represent statistical noise prior to embedding along the non-observable space of the courier variables, i.e., the noisy DOFs as seen in Figure 6. Here, it is observed that the original components along the non-observable space are random and noisy, validating the assumption that the non-observable space represents noise. Upon embedding, the components are replaced by the permuted evidence-based records while preserving statistical properties of the components such as the mean and variance. To verify, the basis matrix of the non-observable space $\mathbf{Q}^{\perp}$ of one subsystem is used to extract the obfuscated records and the one-time-pad is used to reverse the process, yielding the embedded temporal information of the other subsystem as seen in Figure 7. It is observed the embedding is

recovered with perfect accuracy, implying that any minor distortion in the process variables can be detected.

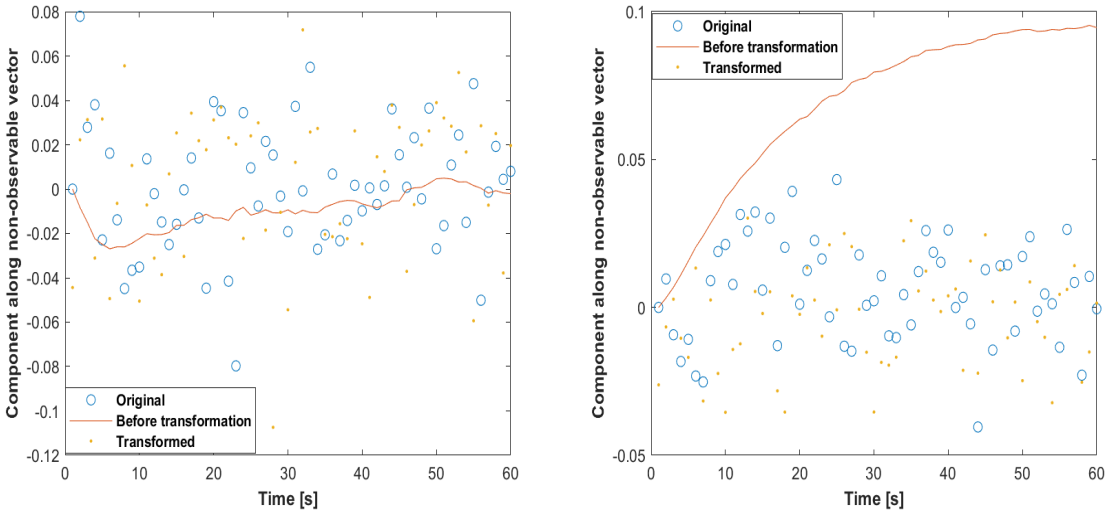

Figure 6: Non-observable space: noisy DOFs

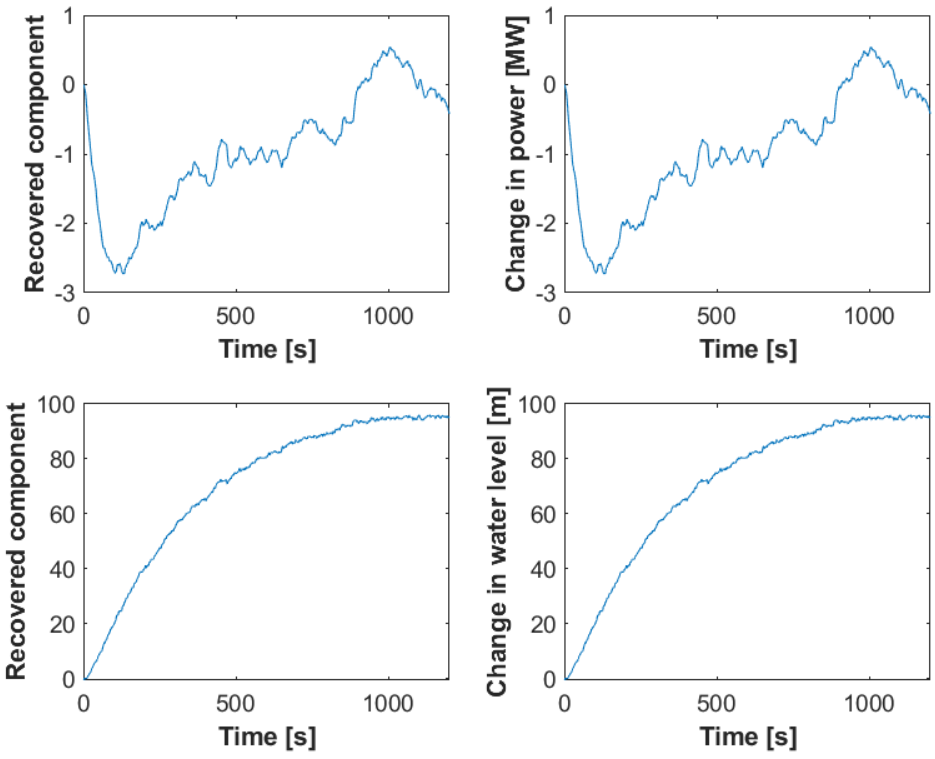

Figure 7: Recovered evidence-based records in steam generator (top-left) and core (bottom-left); source of evidence-based record in core (top-right) and steam-generator (bottom-right)

In the case of malicious intrusion via a clever data deception attack that bypasses model-based defenses, the C2 embedding serves as an additional line of defense that can deterministically pinpoint intrusion since it is highly improbable that the data deception attack preserves the embedded evidence-based records in the non-observable space of the process variables. For example, a replay attack that falsifies data by replaying previously authenticated data to bypass statistical techniques is instantly detected since the embedded information changes with time due to the one-time-pad algorithm. On the other hand, anomalies arising out of external disturbances, equipment degradation etc. may be detected using model-based metrics such as LOCs and HOCs as explored previously in the nuclear community [31]. Here, there is no falsification of the process variables, and the data still carries the C2-embedded evidence-based records. Therefore, the C2 paradigm may be used to differentiate between anomalies arising out of malicious intrusion and natural causes to enable operators/owners take appropriate measures.

3.3 Statistical Validation

It is expected that a successful implementation of the C2 paradigm results in a system response (states, inputs, outputs etc.) that is identical to normal operation within the statistical noise expected in the system. In this section, the C2 paradigm is first validated visually via histograms and plots of the states, inputs, outputs, residuals, and the controller cost. Subsequently, hypothesis tests are performed on the residual statistics based on derivations from control theory using the Kalman filter. The following subsections provide a background of the measures used followed by the validation results obtained.

3.3.1 Background

The Kalman filter [97], [98], developed in the control community, provides an optimal estimate of responses in a linear sense, and for the given LTI problem, the statistical distribution of the residuals $\mathbf{z}_A$ may be pre-computed. Here, the residuals are calculated as the difference between the observed output and the estimated output, which is expected to approach a zero-mean Gaussian distribution of some variance that can be computed using the variances of the process and measurement noise, and the system matrices under conditions of stability and observability. The filter may be implemented recursively in a manner that updates estimates of the covariances

as more measurements appear, converging to a steady-state value depending on the stability of the state matrix. In the discrete case, the steady-state covariance matrix $\mathbf{S}_\mathrm{A}$ of the residual of the core, also called innovation in the control community, is given by the expression $\mathbf{S}_\mathrm{A} = \mathbf{R}_\mathrm{A} + \mathbf{H}_\mathrm{A}\mathbf{P}_\mathrm{A}\mathbf{H}_\mathrm{A}^\mathrm{T}$ where the Lyapunov equation $\mathbf{P}_\mathrm{A} = \mathbf{F}_\mathrm{A}\mathbf{P}_\mathrm{A}\mathbf{F}_\mathrm{A} + \mathbf{Q}_\mathrm{A}$ must be solved to obtain the steady-state error covariance matrix $\mathbf{P}_\mathrm{A}$ using the process noise covariance matrix $\mathbf{Q}_\mathrm{A}$.

In model-based defenses, the control system is often equipped with a χ^2-detector [98] that analyzes the distribution of residuals. Since the residual is expected to be normally distributed, a validation gate may be setup by computing $\mathbf{z}_\mathrm{A}\mathbf{S}_\mathrm{A}^{-1}\mathbf{z}_\mathrm{A}^\mathrm{T} \leq g^2$ for some threshold quantity g. Although the C2 paradigm is designed to protect the system and not evade detectors, a successful covert implementation of the C2 paradigm requires the distributions of the residuals to be preserved to render it resistant to AI/ML and statistical techniques. In other words, the perturbations along the non-observable space of the state estimates must not change the statistical distribution of the residuals characterized by zero mean and the precomputed variance mentioned in the previous paragraph. While histograms are useful in analyzing distributions and providing visual validation, the Kolmogorov-Smirnov test is a powerful statistical test that determines whether two samples arise from the same distribution at the required significance level. In the context of residual comparison, under the null hypothesis, it is assumed that the two different sets of residuals arise from the same normal distribution while the alternate hypothesis indicates otherwise.

3.3.2 Results

As seen in Figure 8, it is observed that the embedding the evidence-based records from the water level of the steam generator along the core of the reactor results in a perturbed response (labeled ‘Modified’) that is similar to the case under normal operation (labeled ‘Original’). Using the Kalman filter, a general trend of the response is obtained to visually validate the expectation that the perturbed response is a noisy version of the original.

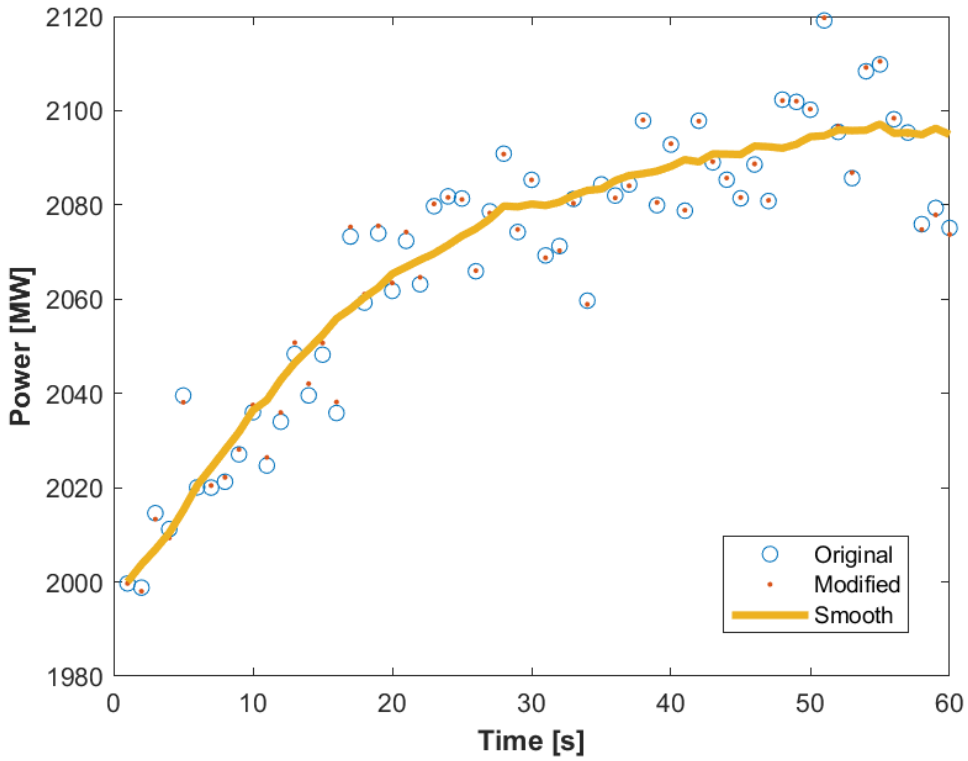

Figure 8: Validation of core response behavior

The residuals of the core response may then be computed using the Kalman filter estimate to ensure that the C2 paradigm is covert to χ^2-detectors and does not artificially inflate the noise in the system. While Figures 9 and 10 serves as visual validation, the Kolmogorov-Smirnov test is used to test whether the residuals are similar in distribution. Under the null hypothesis, it is assumed that the residuals with and without the C2 embedding have the same underlying distribution, and the test fails to reject this null hypothesis at a significance level of 0.05 over multiple runs with p-values > 0.5.

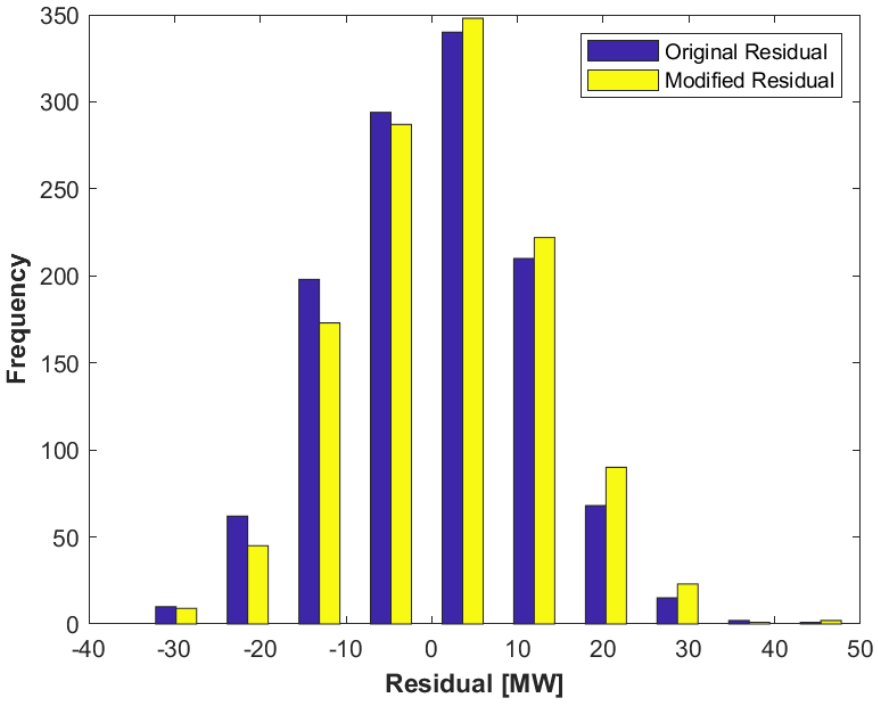

Figure 9: Validation of residual distribution

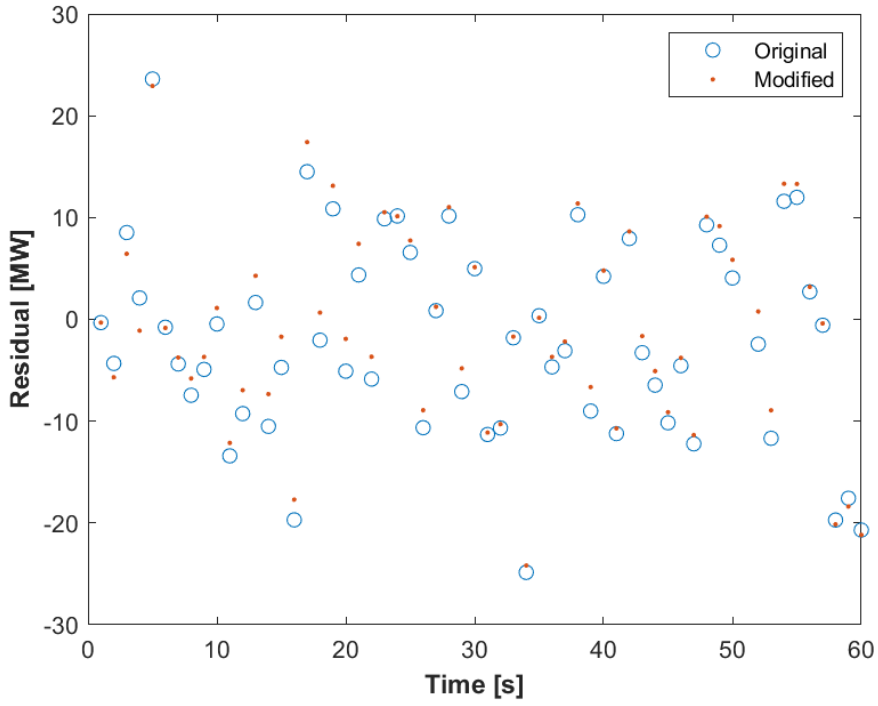

Figure 10: Validation of residuals over time

While the previous results focus on individual responses, it is recognized that the C2 implementation must lie in the non-observable space of the entire system to have zero-impact on the process, i.e., the perturbations must not affect the active DOFs in the system such as the correlations between response variables. For instance, physics indicates that the core power and the fuel temperature are positively correlated with time after an initial delay due to the heat transfer coefficient of the fuel pin. Such correlations must be preserved even with the perturbations with any deviation being the result of measurement noise in the system. As seen in Figure 11, it is observed that these correlations are preserved within statistical noise, serving as additional validation for the C2 implementation.

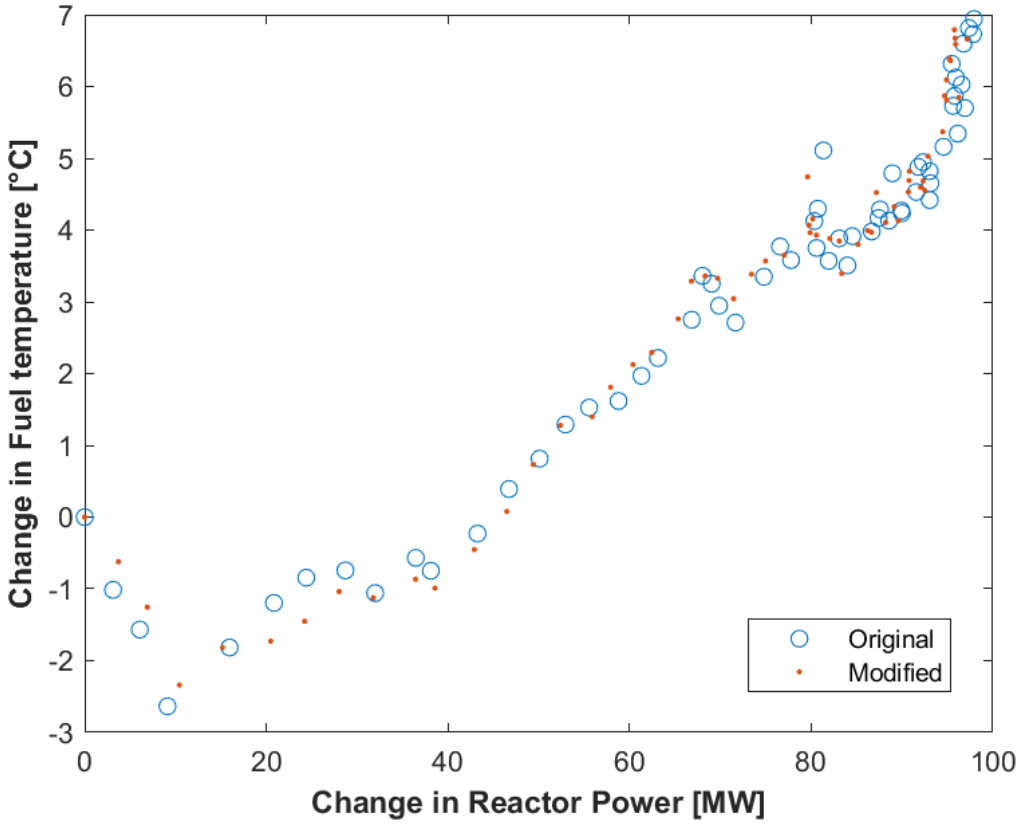

Figure 11: Validation of response correlation

Since the embedding is done along the process variables such as the state estimate and output, it is entirely possible for the system to manipulate the control input to the core, i.e., the inlet temperature, to "make up" or account for these perturbations. Therefore, it is necessary to

validate that the change in the control input is also within noise levels to ensure that the embedding of the evidence-based records from the steam generator truly have zero impact on the process. As seen in Figure 12, the control input after embedding is visually similar to that under normal operation, further validating the C2 implementation.

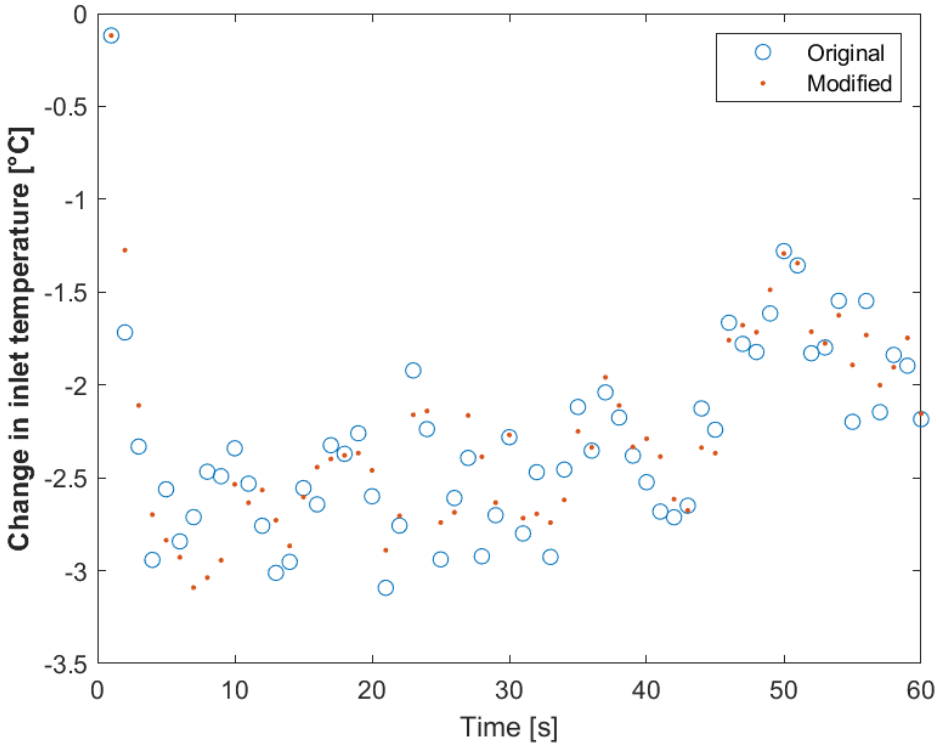

Figure 12: Validation of control input

As a final validation of the zero-impact condition on the control algorithm, the evolution of the state variables and control input with time is summarized using the measure of controller cost. Note that the Kalman filter and Kalman gain algorithms are derived by optimizing the process with respect to the cost. If the operation of the system with and without the C2 implementation is indeed identical, it is expected that the cost of the controller remains the same within statistical noise. As mentioned in Section 1.4, this may be achieved by further constraining the perturbations to be orthogonal to the gradient of the controller cost.

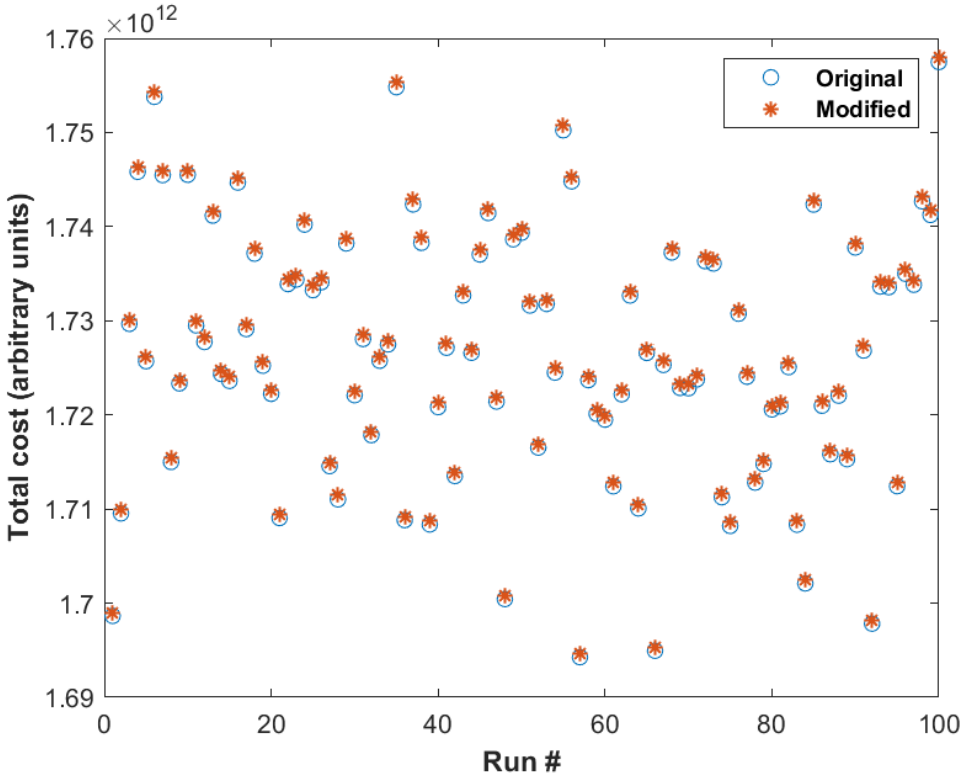

Figure 13: Validation of controller cost

Over multiple runs as seen in Figure 13, it is observed that the controller cost before and after the embedding is identical with any variation being much smaller than the variance due to noise. Thus, the zero-impact and zero-observability constraints of the C2 paradigm have been statistically validated for the simple reactor case.

In recent years, adversaries have augmented their analysis of industrial systems with AI/ML data mining and pattern discovery tools to uncover underlying correlations and exploit vulnerabilities in the model. The statistical methods presented in this chapter analyze the temporal evolution indirectly through cost-function and residual analysis. However, the time-dependence of the response must be analyzed directly using features of the data to ensure that the C2 embedding does not affect dominant features or induce new correlations. These features are typically extracted using AI/ML tools such as long short-term memory (LSTM) networks and convolutional neural networks (CNNs). Therefore, in addition to the statistical analysis above, the C2 implementation must be validated against AI/ML tools to ensure that the zero-observability and zero-impact constraints are satisfied. The following section evaluates the efficacy of the C2 paradigm against

AI/ML tools using the GAN for Anomaly Detection (GAN-AD) tool developed by the data science community to secure water treatment plants.

3.4 AI Validation

In this section, the C2 implementation in the linearized reactor model is validated against the GAN-AD AI/ML tool[99]–[102]. For simplicity, only the core model and its output reactor power are used, and the evidence-based records from the output are embedded along the state-estimate of the Kalman filter. The GAN-AD tool consists of two LSTM networks trained against each other using a GAN framework. LSTMs are a recurrent neural network architecture suitable for time-series data due to their ability to remember/forget sequences and keep track of dependencies within the data. The GAN framework consists of two competing networks, called generator and discriminator respectively, that compete against each other to extract features from the time-series data and learn the true underlying distribution of the process. The generator network attempts to transform random inputs into time-series data representative of the training samples based on feedback from the discriminator. The goal of the discriminator is to distinguish between synthetic samples created by the generator and the true data supplied as the training set. With sufficient training, it is expected that the generator can accurately mimic the features of the time-series data and fool the discriminator into classifying the synthetic samples as authentic. Additionally, the trained discriminator may be utilized as a statistical anomaly detection tool capable of detecting anomalous data that are not representative of the training set. This chapter relies on this observation to validate the C2 paradigm by training a GAN-AD model on operational data from normal operation and using the trained discriminator to attempt to distinguish normal operational data from C2-embedded data. If the discriminator is unable to distinguish, it may be likely that the two datasets come from the same process, which is the goal of the C2 paradigm. The following sections provide a background of LSTMs, GAN, and GAN-AD, followed by a summary of the results from validating C2 against AI.

3.4.1 Background

Long short-term memory networks (LSTMs)

LSTMs [101], [102] are a modification of recurrent neural network architectures that attempt to mitigate the effects of the vanishing gradient problem by passing the gradient information directly to subsequent layers. In doing so, they avoid some of the problems associated with multiplying finite-precision numbers that may go to zero or infinity with time as past information is continually incorporated. Nevertheless, they still retain the ability of recurrent networks to learn and store information over extended time-intervals through feedback connections. A typical LSTM unit consists of three gates, namely, the input gate, the output gate, and the forget gate, that regulate the flow of information through the cell as shown in Figure 14 [103]. The input gate determines the relevant information that must be added at the current time-step, the forget gate determines the relevant information from previous time-steps, and the output state determines the cell state in the next time-step based on this information. The cell state serves as the memory of the network as sequential data is processed and the use of gates ensures that the memory may be extended to longer time-steps than usually feasible with recurrent neural networks. This contrasts from simple feed-forward network architectures that do not have feedback connections to understand the sequential nature of data.

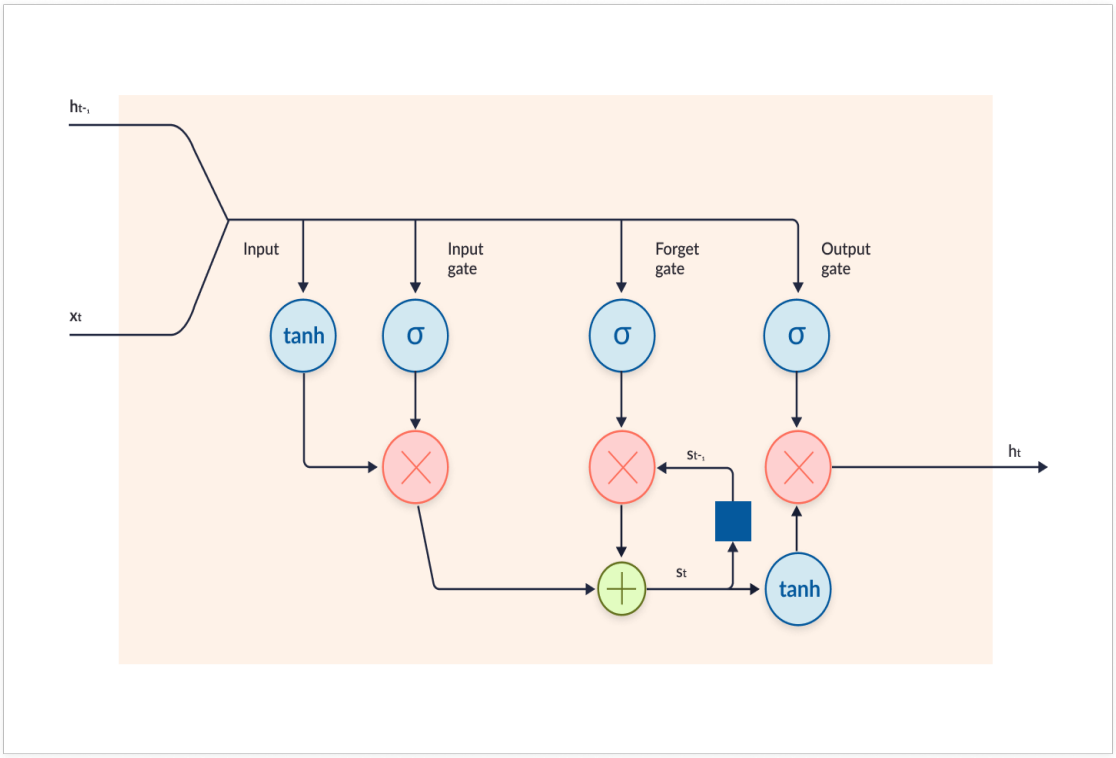

Figure 14: LSTM Architecture

The ability to retain memory and understand sequences makes LSTMs ideal for time-series data. They are widely used in forecasting and classification applications including but not limited to speech recognition, predictive texts, handwriting recognition, music composition, and anomaly detection. For instance, LSTMs were used for speech recognition in Google Voice, Polly for Alexa, and to aid in automatic translations for Facebook. One of the most significant achievements, however, came in their implementation in the OpenAI bots for the video game *Dota 2* [26], where the five-man bot team defeated the then world champions, OG, under limited conditions through clever use of in-game sustenance and co-ordination strategies that did not just rely on superior reflexes inherent in computers. Remarkably, the bots were trained from scratch through hundreds of thousands of hours of repeated games against themselves in an unsupervised manner, i.e., no footage of human professionals was used as training data. Similar progress was achieved in *Starcraft II* with DeepMind's program AlphaStar [104].

In industry, LSTMs have been utilized for predictive maintenance, anomaly detection, and business process monitoring applications [102], [105], [106]. For example, time-series data from the process variables of a jet engine was used to train an LSTM model to predict the remaining

useful life of the system. These predictions are useful in detecting incipient signs of failure for maintenance purposes and can save resources otherwise lost due to system downtime, replacement costs, security concerns etc. In business settings, logs of completed processes may be used to predict the runtime of existing processes and make informed decisions for scheduling and optimization purposes. The sequential nature of the data makes LSTMs a lucrative process monitoring tool and research has demonstrated their ability to outperform other predictive models without the need for excessive tuning to the target application. Lastly, as discussed in greater detail in subsequent sections, LSTMs may be used to extract and learn features from sensor data in systems for anomaly detection. These anomalies may be a result of cyberattacks, equipment degradation, external disturbances etc. One such tool, GAN-AD, is chosen as the representative AI/ML tool to validate the C2 paradigm in this manuscript.

Generative Adversarial Networks (GANs)

GANs, introduced by Goodfellow et al. [14], are a recent milestone development in AI/ML research widely using in image processing, synthetic data generation, and advertising. The GAN framework consists of a generator and a discriminator that compete against each other in a zero-sum game where the goal of the generator is to learn the feature space of the input and create synthetic samples while the discriminator attempts to distinguish between the training dataset and the synthetic samples. As shown in Figure 15 [107], the generator is penalized while the discriminator is rewarded if the latter is able to successfully distinguish between the data and vice versa otherwise. The input to the generator is typically random noise from a known multivariate statistical distribution, and the generator transforms this into data representative of the training set using a series of nonlinear transformations depending on the generator architecture. The output is then fed into the discriminator along with samples from the input training data, which then classifies the data as synthetic or original and the resulting loss is fed back into the generator for further training. In the Wasserstein loss function for example, given a real training sample $\mathbf{x}$, noisy input $\mathbf{z}$, generator function $G(\cdot)$ and discriminator function $D(\cdot)$, the discriminator seeks to maximize the loss $D(\mathbf{x}) - D(G(\mathbf{z}))$ while the generator seeks to maximize $D(G(\mathbf{z}))$. Intuitively, this implies that the discriminator is judged by its ability to classify real samples as real and synthetic samples as fake, whereas the generator is judged by synthetic samples that are misclassified as real by the discriminator.

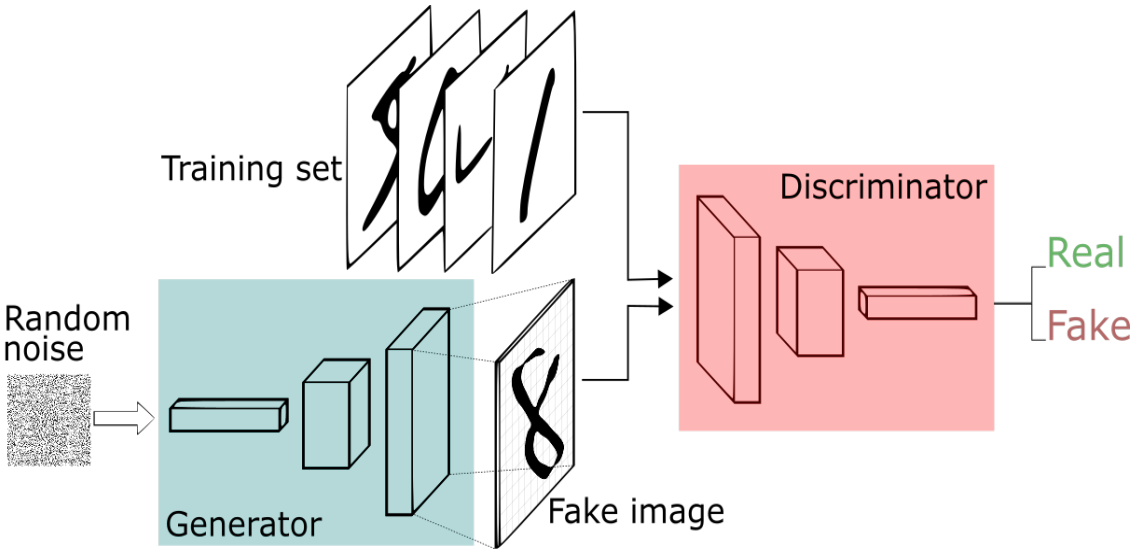

Figure 15: GAN Architecture

GAN training typically occurs in alternating periods where either the generator or the discriminator weights are fixed while the other trains for one or more epochs. It is difficult to ascertain if and when convergence occurs during the training process and the training periods must be carefully juggled until the two competing networks are sufficiently trained. For example, if a generator is extremely good at generating representative samples, the discriminator cannot guess better than random leading to meaningless outputs. However, since these outputs are fed back into the generator via the loss function, the generator effectively trains in subsequent epochs on meaningless data, which may degrade the performance of the generator. Another issue is that of mode collapse, where the generator fails to generalize the training data and instead focuses on a subset of features that are still "valid" but not entirely representative. For instance, given a training set on the digits 0-9, the generator may only learn to generate ones and threes, which are satisfactory to the discriminator and thus no further training occurs. To alleviate this issue, mutual information-based loss functions have been proposed where the mutual information itself is estimated using a supporting neural network, thus forcing the generator to create synthetic samples representative of the entire dataset as opposed to only a subset. Common examples of loss functions involve measures that compute the difference in probability density functions of the input data such as Wasserstein loss, minimax loss etc.

The generator and discriminator architecture vary depending on the application and the input data. Convolutional neural networks (CNNs) have enjoyed great success in image processing

applications and are widely used in the generation of deepfakes, AI art [5], [24] etc. The generator is typically a deconvolutional neural network that projects the relatively low-dimensional noise input to a high-dimensional image while the discriminator is a CNN that reduces the image to a single number describing its "authenticity". Additionally, a combination of DNNs, RL, and the GAN framework have been used in the medical industry for AI-inspired drug discovery [108]. Here, the AI predictive model scours through existing drug databases, attempts to learn common relationships and drug properties, and synthesizes new candidate drug models to significantly speed up drug research. While the area of AI drug discovery is still in its infancy, the model by Insilico Medicine has shown great promise with successful tests in mice using some of the artificially generated drugs. With regards to time-series applications, both the generator and discriminator may be LSTMs that are well-suited in learning sequences and correlations over time intervals. These may be leveraged for synthetic data generation to create energy portfolios for optimization and anomaly detection to perform predictive maintenance, detect intrusion etc. The GAN-AD tool outlined in the next section is one such application of LSTMs trained using the GAN framework for anomaly detection.

Generative Adversarial Networks-based Anomaly Detection (GAN-AD)

The GAN-AD model, later published as Multivariate Anomaly Detection with GAN (MAD-GAN) [99], [100], is a state-of-the-art anomaly detection tool developed by the machine learning community for complex CPS. The architecture, illustrated in Figure 16, pits two LSTMs against each other in a GAN framework to learn the underlying distribution of sensor and actuator data in the CPS. Since sensors and actuators are highly correlated, the dimensionality of the time-series data may be modeled using substantially fewer variables by the generator. The discriminator attempts to differentiate between the synthetic samples from the generator and real samples from data obtained from a complex six-stage water treatment plant. The model is then trained on the water treatment data until the generator can sufficiently model the underlying probability distribution of the sensors and actuators.

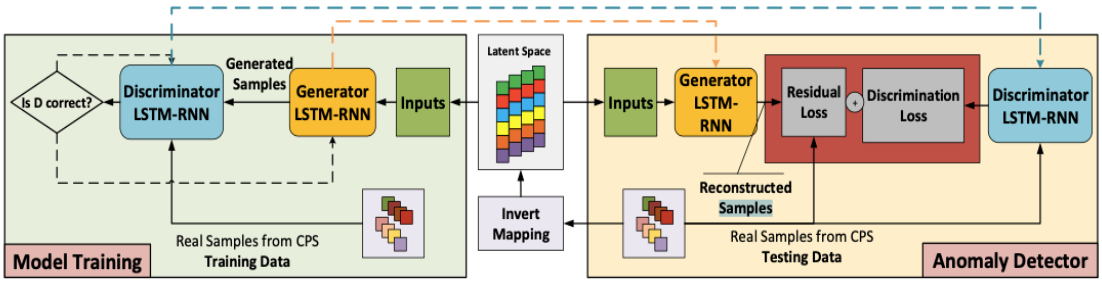

Figure 16: GAN-AD/MAD-GAN Architecture

While the generator typically maps from the latent space (inputs) to the sample space (sensor data), an inverse mapping from the sample space to the latent space is found by the authors through similarity-based iterative methods. Given the inverse mapping and the ability of the trained discriminator to detect differences in probability distributions, an anomaly score metric is derived for the purposes of anomaly detection. The time-series data may be passed to the detector for classification as anomalous or otherwise in a pointwise or sample-wise (collection of consecutive points) manner. The ability of the detector is enhanced when multiple responses are incorporated since the correlations among them often provide additional insight into system behavior that cannot be gleaned from a single response. Note that such detectors are still passive model-based methods and may be easily bypassed by falsification attacks such as the replay attack popularized by the Stuxnet virus that falsifies process variables using authentic data from previous time-steps. Nevertheless, the GAN-AD model provides a powerful validation tool to determine if the C2 paradigm truly satisfies zero-observability and zero-impact constraints by testing for discrepancies in the distribution of the process variables with and without the embedding. This represents the ultimate test for the resilience of C2 to AI/ML techniques, as discussed section 3.4.2.

Isolation Forest

Isolation forest (iForest) [109] is a state-of-the-art unsupervised anomaly-detection tool that takes a different approach to most anomaly detector techniques. While methods such as one-class SVM attempt to profile normal instances, bound them as tightly as possible using a hypersphere, and classify any outliers as anomalous, iForest profiles the anomalies themselves. The underlying argument is that anomalies are far and few and unique in their occurrence, and consequently must be easier to "isolate" from a given dataset than other points. The dataset is split

using binary trees by randomly selecting a feature and a random threshold to partition the data. On average, anomalous points require a shorter path length, i.e., fewer splits, to be isolated than their normal counterparts, and each point can be assigned an anomaly score using this metric averaged over multiple trees as shown in Figure 17.

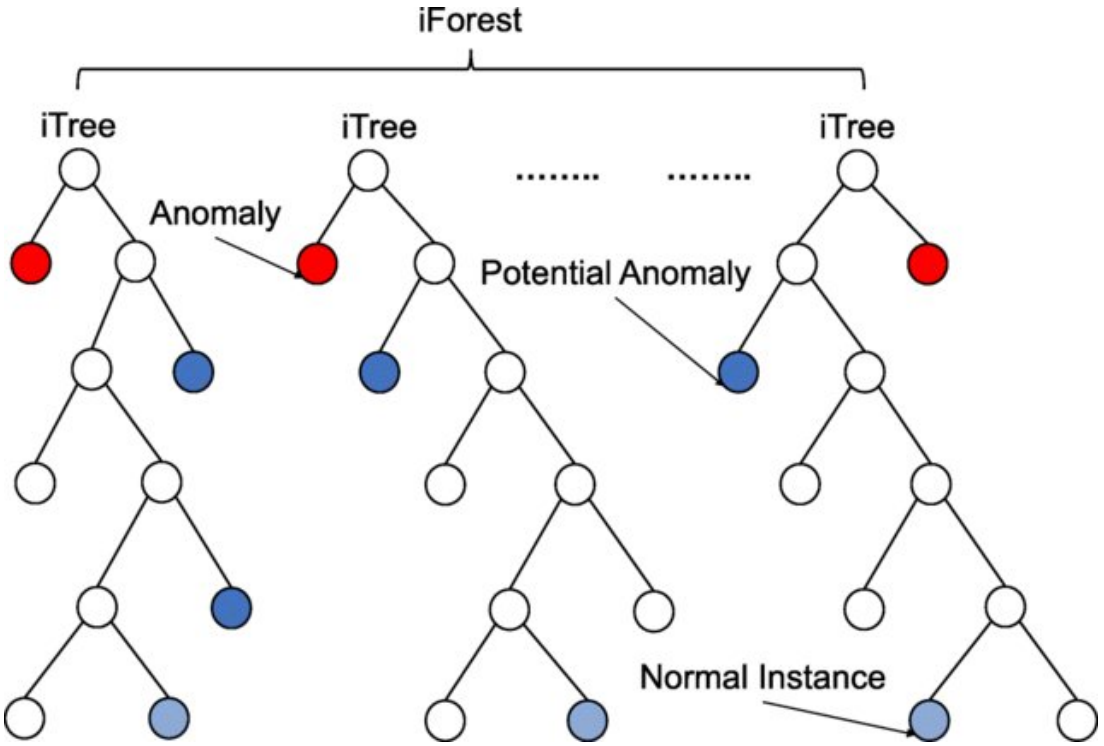

Figure 17: Isolation Forest

The C2 paradigm can be validated by looking for point-wise anomalies in the embedded data. If the zero-observability and zero-impact constraints are not satisfied, it is expected that the isolation forest algorithm will be able to classify anomalous points better than random on a large enough dataset with multiple samples. Failure to isolate the C2-embedded points better than random may be taken as evidence that the anomalous points are statistically similar to the non-anomalous entries as demonstrated in the following section. This is the core of the C2 approach, a time-series with and without the embedding must appear statistically similar in their deterministic as well as stochastic components.

3.4.2 Results

In this numerical experiment, the linearized core model is executed multiple times without the C2 physical process defense using the Dymola simulation software [110], [111] to generate a dataset representative of normal operation. The GAN-AD tool is trained on this dataset until it learns the underlying distribution of the input data and the generator can reproduce representative samples that bypass the discriminator. Once trained, the discriminator is isolated from the architecture to serve as an anomaly detection tool. Multiple datasets are generated with the C2 defense embedded into the process variables and passed to the discriminator as testing data to test its ability to distinguish between the C2-embedded data and the data under normal operation. This represents the ultimate test of the C2 paradigm since the discriminator was only trained on normal operational data. In other words, the C2-embedded data was unseen by the discriminator and consequently, the latter must be able to detect any deviations in the feature-space caused by the C2 embedding. If the discriminator is unable to perform better than random, it is likely that the distribution and feature-space of both the datasets are identical, fulfilling the goals of the C2 paradigm.

In this experiment, the same neural network parameters and preprocessing criteria are used on the training data as in the original GAN-AD manuscript. The GAN-AD model is trained as a two-player minimax game with the discriminator loss and generator loss to be minimized as provided in Eqs. 3.6 and 3.7. Here, the same notation for the input noise, generator, discriminator, and samples is used as in the background on GANs in section 3.4.1 where m denotes the number of samples in the dataset. The LSTM generator consists of 3 layers with 100 hidden units each while the discriminator consists of 1 layer with 100 hidden units. The input latent space to the generator is 15-dimensional to learn features from the temporal data.

$$D_{loss} = \min \frac{1}{m} \sum_{i=1}^{m} \left[-\log D(\mathbf{x}_i) - \log \left(1 - D\left(G(\mathbf{z}_i) \right) \right) \right] \tag{3.6}$$

$$G_{loss} = \min \frac{1}{m} \sum_{i=1}^{m} \log \left(-D\left(G(\mathbf{z}_i) \right) \right) \tag{3.7}$$

The training input is the response **y** of the reactor, representing the power level at steady-state conditions. The data was collected at time-intervals of 1 second for a total duration of 496,800

seconds (5.75 days). The GAN-AD is trained using two types of datasets, namely, pointwise and sample-wise. With pointwise data, the GAN-AD model attempts to distinguish individual time-steps in the output data as anomalous or otherwise, whereas with sample-wise data, the GAN-AD model learns temporal features from a time-series of length 120 time-steps (2 mins) and the discriminator attempts to classify the given test sequence as anomalous or otherwise. In this numerical experiment, equal number of normal operation and C2-embedded datasets were provided to the trained discriminator for the purposes of anomaly detection. The anomaly score metric based on the discriminator output and the generator reconstruction error was adopted from the GAN-AD model to test the C2 paradigm. A given point/sample was declared as anomalous if its anomaly score was above a certain threshold value τ which may be varied based on the desired true positive and false positive rates.

As mentioned earlier, the GAN-AD model is trained on data obtained from normal operation and tested on C2-embedded data to assess differences in the statistical properties and the underlying probability distribution. Since the problem is a binary classification problem, four metrics may be used to summarize the results of the testing algorithm, namely, the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Here, TP denote the number of C2-embedded samples that were classified as embedded by the discriminator, FP denote the number of samples from normal operation that were classified as embedded, TN denote the number of samples from normal operation that were classified as normal, and lastly, FN denote the number of C2-embedded samples that were classified as normal. Using these metrics, four measures may be derived using the definitions in Eqs. 3.8-3.11.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.10}$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3.11}$$

Typically, the measures of precision and recall are used in the computer science community to describe the performance of a binary classifier. A high precision indicates that the classifier can successfully detect anomalies, while a high recall indicates that the detected anomalies are truly anomalous. As shown in Figures 18 and 19, it can be observed that despite varying the threshold, the GAN-AD model is unable to predict better than random regardless of pointwise or sample-wise data. In cases where a high recall is obtained, the false positive rate is correspondingly high indicating that the classifier classifies most normal operation and C2-embedded datasets as anomalous. Attempting to decrease the false positive rate results in a low recall, indicating that the classifier mostly classifies both datasets as normal operation. This is better captured via the precision and accuracy metrics which are capped around 50%, the equivalent of randomness/coin-toss probability.

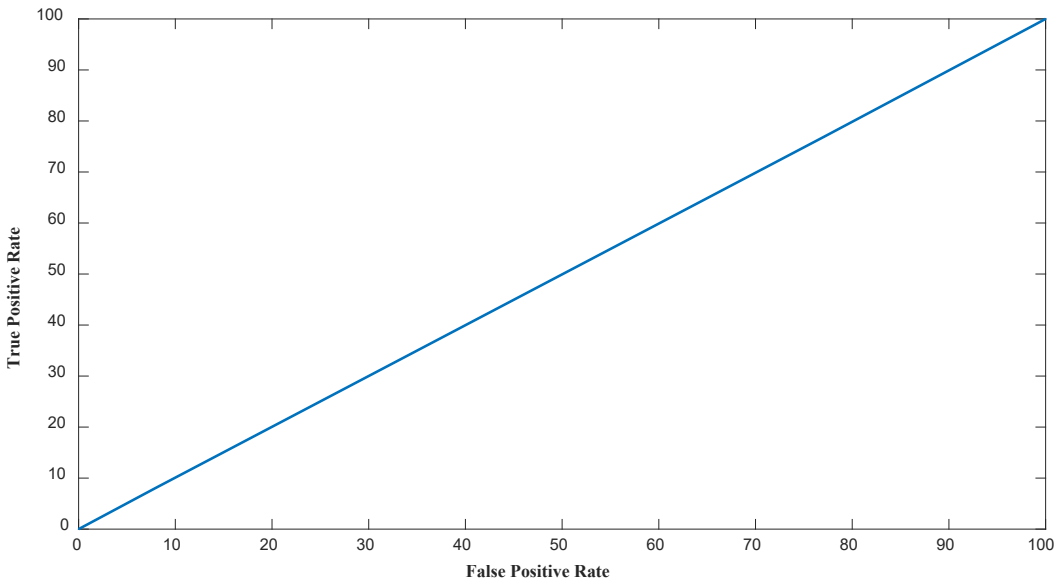

Figure 18: ROC curve for point-wise results

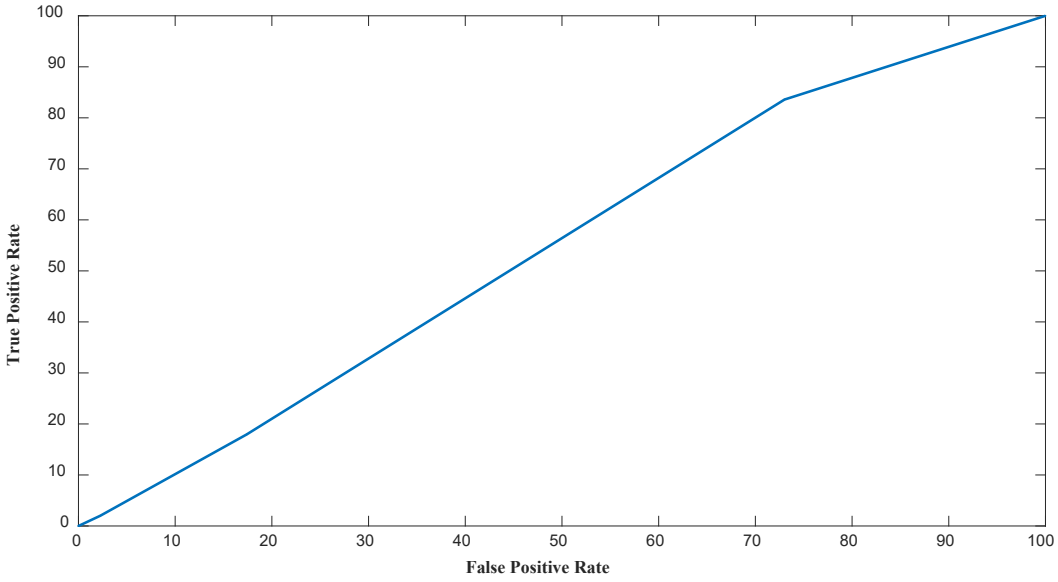

Figure 19: ROC curve for sample-wise results

The above results indicate that the statistical distribution of the system output with and without the C2 embedding is similar, as validated visually earlier with the histograms in section 3.3.2. As mentioned in the introductory section of this chapter, this implies that the C2 implementation is resistant to pattern discovery by AI/ML techniques since the temporal features extracted from the C2-embedded data are identical to the normal operation case. Combined with the statistical results in section 3.3.2, it is demonstrated that the zero-impact and zero-observability constraints are satisfied.

The iForest numerical experiment was conducted on the above dataset with the same preprocessing parameters and without the label information. However, for the sake of analysis, the iForest labels were compared to the true labels to determine if there were any discrepancies in the point-wise data. Half the data was contaminated with anomalies (the C2 embedding) at every alternate time-step, and a set of 100 binary trees were used to extract features and isolate the anomalies. The iForest algorithm was implemented using the PyOD library [112], a state-of-the-art anomaly detection library for time-series data.

Table 1: Point-wise results from iForest

Metric	Score (%)
True Positive Rate	49.93
False Positive Rate	50.08
Accuracy	49.93

It can be seen from Table 1 above that the iForest algorithm is unable to detect the embedded time instances better than random, further validating the claim that the embedding is in line with the principles of zero-observability and zero-impact on the process. Additionally, it can be proven mathematically under assumptions of additive Gaussian noise and working in the non-observable space that the embedding process generates a time-series that has the same deterministic component as the original time-series and the same statistics on the stochastic component. Effectively, the embedding transforms a given time-series into a different realization of the time-series by only manipulating the noise while preserving its statistics. Furthermore, the assumptions of Gaussianty may be relaxed to encompass the family of stable distributions although this is rarely encountered in practice with process noise. In practical applications, different non-Gaussian sources of randomness typically sum up to a Gaussian distribution by virtue of the central limit theorem, and the non-observable space is heavily dominated by noise for any minor trends to be discernible.

To summarize, this chapter demonstrates a simple implementation of the C2 paradigm in a linearized reactor model equipped with a core and steam-generator. Cross-cognizance is induced among the two subsystems using the mathematical framework developed in section 2.3 to serve as an additional layer of physical process defense. It is observed that the given implementation can be used to detect falsification of process variables and can therefore distinguish between anomalies arising out of natural causes and malicious intrusion detection. This fills a key gap left by existing defenses that may be bypassed by replay attacks that preserve statistical similarity and insiders that can detect the additional footprint of active techniques.

The work has several extensions. First, it is recognized that the above implementation was intended as proof-of-concept and is limited in its scope due to the simple reactor model under consideration which only had a few degrees of freedom for the C2 embedding. Therefore, the work

must be extended to a real representative model with multiple sensors and nonlinear dynamics as encountered in the physical world. The proposed work seeks to accomplish this using the open-source TRANSFORM Library [113] developed by Oak Ridge National Lab for the modeling software Dymola. Secondly, it is observed that one subsystem may be used to recover lost data in the case of data falsification since it contains the necessary information to retrieve and restore the falsified process variables of the other subsystem. This is because the evidence-based records are created based on the temporal evolution of the process variables prior to falsification by an adversary. Thus, the C2 paradigm serves a dual purpose of intrusion detection and data recovery to ensure robust operation of the system even under compromised environments, representing the ultimate goal of OT defenses. Lastly, with the advent of automation and remote operation, it is critical to develop an automated method of intrusion detection and data recovery for critical infrastructure to continue operation in inaccessible environments. This is explored through the development of automated C2 modules in Dymola for a representative small modular reactor using the TRANSFORM package in chapter 4.

4. CASE STUDY: DATA RECOVERY

This chapter concerns the development of automated C2 modules using the TRANSFORM library in Dymola for a representative small modular reactor and borrows content published in the journal [114], Progress in Nuclear Energy. One of the important premises of unattended operation, a highly promoted characteristic of fission batteries and advanced microreactors, is the ability to automate the analysis of sensors data used in support of operational monitoring and control. To meet this vision, this work proposes a new monitoring and data recovery paradigm to ensure resilience against data corruption which may be the result of malicious intrusion into the reactor operational network. This is paramount to ensure 100% availability under contingency scenarios such as cyberattacks. In support of this vision, earlier work has presented the concept of covert cognizance and demonstrated its mathematical ability to identify and embed cognizance parameters under the noise-dominated null space of the sensors data. This work extends this concept and applies it in real-time to demonstrate three key characteristics: zero-impact, zero-observability, and data recovery, where the first characteristic is to ensure no impact on operation, the second is immunity to discovery by pattern recognition techniques, and the third is to allow recovery of corrupt or falsified data. Recognizing that fission batteries are designed to operate under steady state most of the time, we elect to employ a small modular reactor model under transient operational conditions to demonstrate the operational resilience enabled by the covert cognizance paradigm. Specifically, the PI controller is augmented with the covert cognizance modules to develop self-awareness and enable automatic data recovery. The developed modules are expected to be equally applicable to a wide range of advanced reactor technologies relying on full or partial unattended control.

4.1 Background

Energy resilience, as defined in the FY2018 National Defense Authorization Act under 10 USC § 101(e)(6), refers to “the ability to avoid or prepare for, minimize, adapt to, and recover from anticipated and unanticipated energy disruptions in order to ensure energy availability and resiliency sufficient to provide for mission assurance and readiness, including task critical assets and other mission essential operations related to readiness and to execute and rapidly reestablish

mission essential requirements". In the context of power grids, energy resilience refers to the ability of the power grid to continue delivering energy to critical loads and reduce outages via self-healing and reconfiguration in the event of natural disturbances or malicious cyberattacks. This implies that backup sources of power are needed to maintain power to critical loads and minimize loss of life and/or property during unforeseen events [115].

Along the lines of energy resiliency, the Fission Battery Initiative by Idaho National Lab [116] seeks to develop ready-to-implement battery-like nuclear reactors with the vision of providing economical, reliable, and unattended power to various systems. A few key characteristics of such batteries include prompt installation, standardized sizes, and the ability to be fault-tolerant and achieve 100% availability under different operating conditions. To achieve this vision, testbeds such as MARVEL [117] have been proposed for proof-of-concept demonstration and the validation of fission battery technologies. In the upcoming years, these testbeds are expected to be equipped with digital twin technologies to achieve secure, robust and unattended operation with a focus on automated intrusion detection and data recovery technologies. This is especially relevant in the face of cyberthreats that may seek to disrupt small-scale remote systems such as decentralized microgrids operating in island mode.

The demand for automated detection and data recovery at the process data level in the face of cyberthreats is largely unmet when considering knowledgeable adversaries such as insiders and advanced persistent threat actors that have the technical know-how to evade statistical model-based measures and can manipulate the system to undesirable states in subtle ways [11,12]. To this end, the present work seeks to address this challenge using a method referred to as covert cognizance (C^2) [13,14], which is a novel active defense paradigm that protects a system at the process level as a safeguard against insiders that have bypassed existing information technology (IT) measures and passive operational technology (OT) measures. By augmenting the reactor with automated C^2 modules, self-awareness is induced across the system sensors to carry information about each other. In the event that a sensor is compromised, the modules serve as an intrusion detection tool that pinpoint the sensor and time of intrusion, ultimately recovering the lost information from other sensors. These design goals reflect the overall goal of OT defenses, energy resiliency and the Fission Battery Initiative– to protect the system at a process level and maintain the functional expectations of a physical process during cyber incidents. Additionally, the demonstration in this work is expected to pave the way for demonstration on testbeds such as

MARVEL and implementation in microreactors and other next generation reactor technologies [118].

The scope of this work is confined to enabling autonomous operation and 100% availability of fission batteries in the face of cyberattacks. Since control strategies, state estimation, verification and validation, and operating conditions for fission batteries are part of ongoing research, the focus of this manuscript is to develop C^2 modules that can easily adapt to conditions and may be readily deployed in any system. The modules are designed to be flexible in their deployment; they may be the software component of embedded firmware, software augmentation of digital twins, or as hardware add-ons to fission battery technologies. They are also agnostic to performance indicators since by definition of the zero-impact and zero-observability criteria, they are designed to not affect the operational data in any meaningful way. In other words, any variations due to C^2 are indistinguishable from natural variations due to system noise by virtue of the C^2 embedding occurring in the so-called non-observable space.

The present work is organized as follows: Section 3 provides a brief background of fission battery technology, the role of automation, and a comprehensive literature review on the existing IT/OT security measures in place. Section 4 introduces the simulated SMR in Dymola, the automated C^2 modules, and the proposed mechanism for intrusion detection and recovery. Section 5 describes the simulation results and validates the implementation using the zero-observability and zero-impact conditions of the C^2 paradigm. Lastly, Section 6 summarizes the results and discusses avenues for future work using the designed C^2 modules.

In the U.S., the microreactor project was proposed to provide small transportable reactor modules to provide ~ 20 MW of power for remote sites, emergency operations, military installations, and space applications. They are expected to operate at temperatures exceeding 600 °C, achieving thermal efficiencies on the order of 32% that rival and even exceed the performance of many large-scale reactors. Owing to their small size, they require substantially less capital than full-scale power plants, have fewer electrical components, and possess a significantly simplified configuration. This makes them ideal for fission-battery-type deployment, as addressed by the fission battery initiative [116], where nuclear energy is miniaturized for plug-n-play deployment with a focus on unattended operation. However, microreactors lose a significant economic advantage due to their small power output, especially in comparison to competing

distributed energy resources of similar power outputs such as distributed solar/wind and battery storage-type technologies.

Automation has been proposed to reduce operations and maintenance (O&M) costs via advanced displays, computer-based control procedures, advanced alarm systems, and computerized operator support systems [119]. The modern instrumentation, controls, and human-machine interface (ICHMI) is one such example that demonstrates the transition from traditional analog systems to digital and potentially automated systems. However, there are a few challenges with the modern ICHMI in microreactors due to the drastically different needs of the latter from conventional reactors [120]. For example, direct sensing capabilities to obtain critical measurements are typically diminished due to the small size and harsh environment of the microreactor, thus increasing the uncertainties built into the safety margins of the reactor. Furthermore, many systems are often shared among the various components of the reactor which requires operators to consider the coupling of dynamics between these systems that they may be unfamiliar with during their experience with conventional reactors [121]. Modern designs have also incorporated the benefits of remote access, and automation technology via the ICHMI to further economize the microreactor. Intelligent control, for example, has been proposed to maintain reliability in adverse scenarios by allowing the reactor to adapt its internal configuration depending on structural changes. In recent years, autonomous frameworks have been developed for microreactors considering their unique challenges to allow for potentially unattended operation for long periods of time [19–22]. With regards to direct application, there is limited research on autonomous control for space systems that have onboard reactors [126]. Simulation models have been created to demonstrate some degree of automation for small modular reactors (SMRs) such as the IRIS module in the TRANSFORM library developed at Oak Ridge National Lab [113]. While microreactors are considered a subset of SMRs, simulation efforts to provide high-fidelity data for microreactors specifically are part of ongoing efforts in the community [127].

While automation technology has become increasingly mature in application domains such as robotics and manufacturing, the technology has not been extensively developed for advanced reactor designs such as microreactors with a focus on robustness, flexibility, and optimization [122]. This is primarily due to the sensitive/proprietary nature of nuclear reactors with some research even showing a negative effect on situational awareness by employees with increased automation [119]. However, it is acknowledged that the conservativeness of the nuclear

community is warranted since the increased connectivity of various components, remote control, and automation open up a new avenue for malicious intruders to access and control the plant. Traditional power plant networks are air-gapped to prevent external access, but attacks such as Stuxnet [68] have shown that even these systems are vulnerable to insiders able to bypass IT protocols, unaccountable employees, and a host of other human factors. Even prior to the discovery of Stuxnet, it was recognized that a highly automated plant control system would have to incorporate diagnostics and response mechanisms for non-standard operational behavior [121].

In response, the control community has attempted to augment and automate intrusion detection using process data analysis (i.e., physics-based detection), and incorporating IT-based measures such as packet analysis. One of the pioneering works [128] in this domain involves the classification of variables into three categories based on their semantics (constants, enums, and continuous variables). This was followed by constructing a behavioral model for each process variable, and raising an alarm when the observed behavior deviated from the expected behavior. Similar work has been done to construct physical model-based and control command analysis-based intrusion detection systems that rely on the knowledge of the physical system, its network topology, and its dynamics to autonomously detect intrusion [27–31].

A common shortcoming of "model-based" automated and passive OT techniques is that they may be bypassed by stealthy attacks launched by knowledgeable adversaries. Such adversaries may be intimately familiar with the system or can attempt to learn system features during an initial waiting period using artificial intelligence/machine-learning (AI/ML) techniques [11,12,32,33]. The battlefield is inherently lopsided in the favor of the attacker with the defender spending far more resources to be bypassed by relatively simple attacks. For instance, a replay attack was shown to be extremely effective in the Stuxnet attack and bypass most passive defensive measures, whereas active solutions proposed to counter replay attacks, e.g., dynamic watermarking [61], sacrifice controller optimality and system performance to enable attack detection. Other active measures such as noise impulse integration [134] is intended for steady-state processes and may have unintended effects during attacks if the target application is sensitive to unmodeled disturbances. Furthermore, while the control community has worked on detection and limiting the impact of stealthy attacks [50], improving the robustness of the system, and discovering zero-day vulnerabilities [36,37], there is a dearth of techniques that can recover data and continue operation during a cyberattack. In other words, mere detection is inadequate for systems such as fission

batteries that are backups to critical infrastructure and must be kept running. Data recovery is still in its infancy with the recent introduction of consensus-based algorithms, redundant sensors, and vault-like data storage mechanisms [38–40].

In fission battery technologies such as microreactors, digital instrumentation and sensors provide measurements that are essential for autonomous control and remote monitoring with no human in the loop under different operating conditions. The control algorithms to achieve autonomous operation must be able to gather information about their operational environment, learn, adapt, anticipate, and take informed control actions. In other words, the monitoring must be supported by intelligent automation and decision-making capabilities with minimal human intervention, i.e., the system must be made self-aware. This includes automated capabilities that can verify the integrity of data in the face of sophisticated cyberattacks and restore the lost data to continue robust operation even if the system is compromised. To address the lack of measures for automated intrusion detection of stealthy attacks and subsequent data recovery, a novel predictive modeling paradigm, C^2[64], was proposed to induce awareness in systems about their operational history. The C^2 paradigm alerts an operator or a digital diagnostics module to stealthy attacks in a deterministic manner by embedding information about the operational history of the process among its process variables. Specifically, the non-observable/noisy space of the process variables are used to carry the information in a random manner since this renders it immune to data mining by humans and AI/ML tools. Additionally, C^2 also permits data recovery in the event that a given sensor is compromised since its information can be recovered using invertible transformations from other sensors. Since the information is in the non-observable space, it was demonstrated in previous work that the methodology does not affect the underlying physical process [65], i.e., a system with C^2 embedded is indistinguishable from one without C^2. The C^2 technology represents the highest level of autonomy [125], [138] achievable by a system where the potential for human intervention is at a minimum. It endows the microreactor with the ability to decide and act autonomously while ignoring operator commands due to a failure in verifying data integrity.

In this work, C^2 modules are created which are capable of automating the above process for a representative SMR using the TRANSFORM package in the Dymola simulation software. The work is intended to validate the C^2 process for SMRs, and to create modules that can be simply attached to any system (plug-n-play) to secure it for other SMR designs, control architectures etc. While our previous work focused on a simplistic linearized reactor model, the present work

implements C^2 in a full-scale non-linear SMR simulation with complex phenomena such as heat transfer, pressure loss etc. The potential for data recovery in a compromised environment is also demonstrated. The goal of this demonstration is to showcase the potential for the plug-n-play modules to be embedded in the control algorithms of microreactors that may be deployed in a battery-like manner. Although not necessarily the intended mode of operation for fission batteries/microreactors, the SMR implementation is chosen to demonstrate that C^2 is also applicable to transient models, i.e., it can exhibit resilience under various operational conditions.

4.2 Integration of C2 with Dymola

The target SMR architecture, the goal of the C^2 modules and their relation to the C^2 paradigm, and a functional description of each component is provided in this section. The International Reactor Innovative and Secure (IRIS) design from the TRANSFORM library in Dymola is chosen as a representative SMR and shown in Figure 1. The SMR is equipped with a control system consisting of feedback controllers, temperature, power and pressure sensors, and actuators that insert reactivity via the control rods and adjust the mass flow rate using a centrifugal pump as shown in Table I.

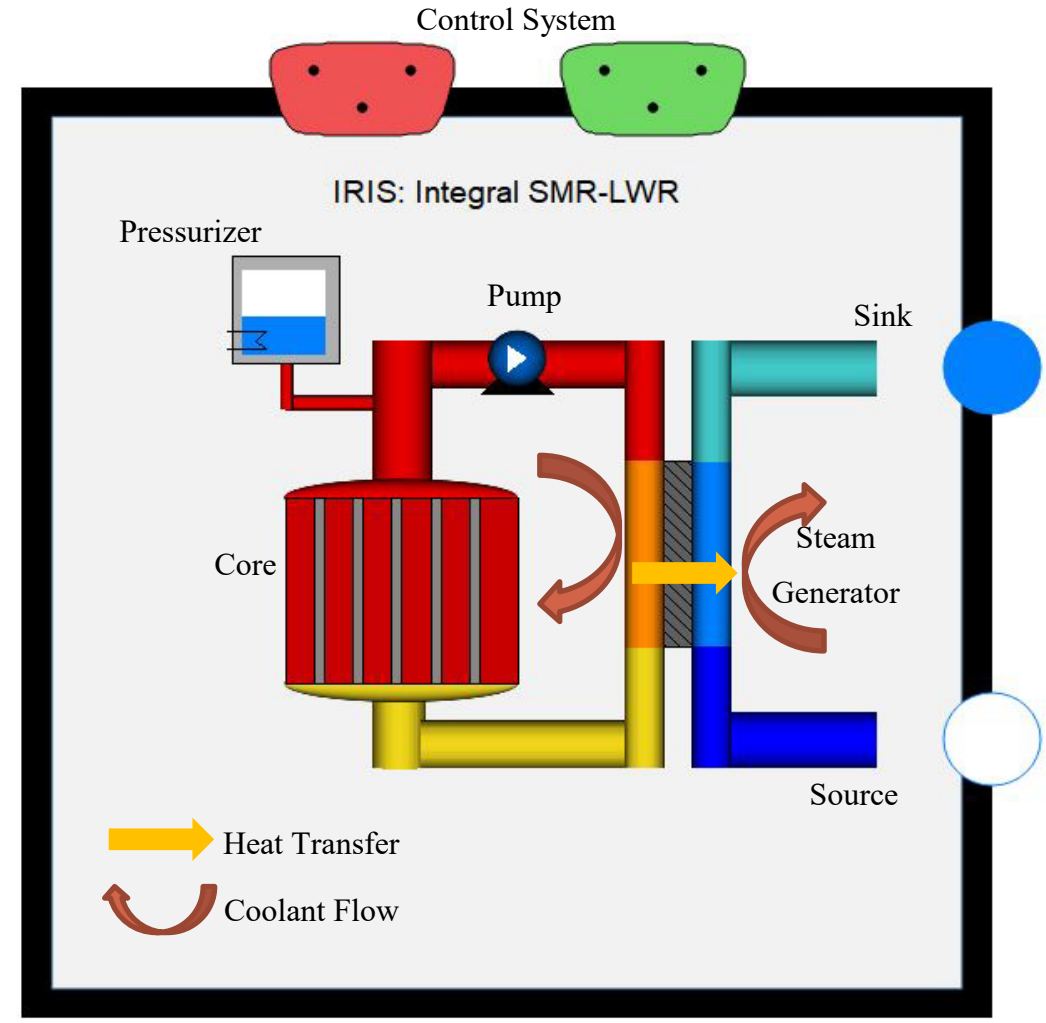

Figure 20: IRIS SMR layout

Table 2: Description of control system

Sensors [Nominal Values]	**Noise Level**	**Corresponding Actuator**
Reactor Power (MW) [1000 MW]	1% noise ~ 5 MW	Control rod reactivity, PI control, limited to $\pm$ 0.01 absolute units
Pressurizer Pressure (bar) [157 bar]	1% noise ~ 1 bar	Heater, constant power, turns on/off depending on pressure
Core Inlet Temperature (°C) [285 °C]	0.1% noise ~ 0.5 °C	Pump speed, PI control, limited to [1150, 1950] rpm
Core Outlet Temperature (°C) [315 °C]	0.1% noise ~ 0.5 °C	
Upper Riser Pressure (bar) [157 bar]	1% noise ~ 1 bar	-
Lower Riser Pressure (bar) [157 bar]	1% noise ~ 1 bar	-
Core Outer Plenum Temperature (°C) [321 °C]	0.1% noise ~ 0.5 °C	-
Steam Generator Outer Plenum Temperature (°C) [285 °C]	0.1% noise ~ 0.5 °C	-
Steam Generator Outlet Pressure (bar) [160 bar]	1% noise ~ 1 bar	-
Downcomer Pressure (bar) [160 bar]	1% noise ~ 1 bar	-

For work described herein, the existing design was modified with additional temperature and pressure sensors and corrupted by additive white Gaussian noise for a more realistic simulation as shown in Table 2. Simulation constraints require the PI controllers to operate in discrete time-steps of 1 second, i.e., sensor inputs are used to calculate the corresponding actuator outputs at a rate of 1 sample-per-second. The following sections formulate the problem, the goal of C^2, and the modules developed to implement C^2 in the IRIS SMR for automated intrusion detection and data recovery.

Intrusion detection is initially demonstrated similar to Chapter 3 by embedding information along the non-observable space of the sensors to satisfy the zero-impact and zero-observability design requirements of C^2. Here, information refers to the operational history of a given sensor that can be used to recover the time-series data if that sensor is compromised. Embedding is randomized using a one-time pad [139] to prevent humans and/or pattern detection tools such as

AI/ML from detecting the presence of the information. For data recovery, the information is derived from the temporal data of the sensors so as to be reconstructed using the non-observable space of the other sensors. Mathematically, the above system can be described using the non-autonomous set of differential equations 1-3 for some time t:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{1}$$

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}, t) \tag{2}$$

$$\mathbf{u} = h(\mathbf{y}, t) \tag{3}$$

Here, $\mathbf{x}$ refers to the reactor state that is hidden and generally not observable, $\mathbf{y}$ refers to the noisy measurements by the sensors, and $\mathbf{u}$ refers to the control inputs of the corresponding actuators described in Table 2. The functions $f(\cdot), g(\cdot)$ and $h(\cdot)$ refer to the evolution of the state, the relationship between the state and the sensors, and the controller logic respectively.

Information from a subset of sensors $\mathbf{y}_i$ shall be embedded across another subset $\mathbf{y}_j$ and the information is given by $\{z_i\}$ for each sensor in $\{y_i\}$ at some time t derived using functions $\{m_i(\cdot)\}$ as shown in Eq. 4. To avoid the presence of patterns, as seen in Eq. 4, the information is further obfuscated, randomized and transformed using functions $\{r_i(\cdot)\}$ prior to embedding along the non-observable space of $\mathbf{y}_j$ causing a change $\Delta\mathbf{y}_j$. Let the non-observable space of $\mathbf{y}_j$ be given by the basis vectors in $\mathbf{U}_j^{\perp}$ respectively. While embedding along the non-observable space satisfies the zero-observability criterion, we must also satisfy the zero-impact criterion as seen in Eq. 6. Mathematically, we have:

$$z_i = m_i(\mathbf{y}_i) \tag{4}$$

$$\mathbf{U}_j^{\perp}\left(\mathbf{y}_j + \Delta\mathbf{y}_j\right) = r_i(z_i) \tag{5}$$

$$h\left(\mathbf{y}_j + \Delta\mathbf{y}_j\right) = h\left(\mathbf{y}_j\right) \tag{6}$$

Note that the two sensor subsets are not mutually exclusive and the present work is just one instantiation of the process. Additionally, the observable and non-observable space may be constructed to exploit correlations across time for a given sensor, the correlations among multiple sensors at a given time, or both. As highlighted in previous work [64], one of the main advantages of complex dynamical systems is their reducibility to few dimensions leaving a vast number of degrees of freedom for the non-observable space that can serve as carrier variables for the information.

With the above problem statement, the process is implemented dynamically by automating the generation of the information, its obfuscation, and the embedding along the non-observable space of the target sensor as seen in Figure 2. Specifically, modules are created that may be implemented in a "plug-n-play" manner to achieve cognizance in the SMR's control system as described further in the following subsections.

4.2.1 Development and Implementation of C^2 modules

This section provides a brief description of the C^2 module seen in Figure 2. The C^2 module consists of three major components: generator, scrambler, and embedder. The sensors $\mathbf{y}_i$ from which the information is to be derived is labeled as the "Source" and connected to the generator module. The sensors $\mathbf{y}_j$, whose non-observable space is where the information is to be embedded is labeled as "Target" and connected to the embedder module. Once the information is extracted, it is scrambled using the scrambler module with the aid of a one-time-pad algorithm. The transformed information is then received by the embedder module, which embeds it along the non-observable space of the target sensors. Since the generation and embedding does not have to happen at every time-step, the system is augmented with two Boolean pulses, "Generate" and "Embed", that signal when the information must be extracted and embedded respectively.

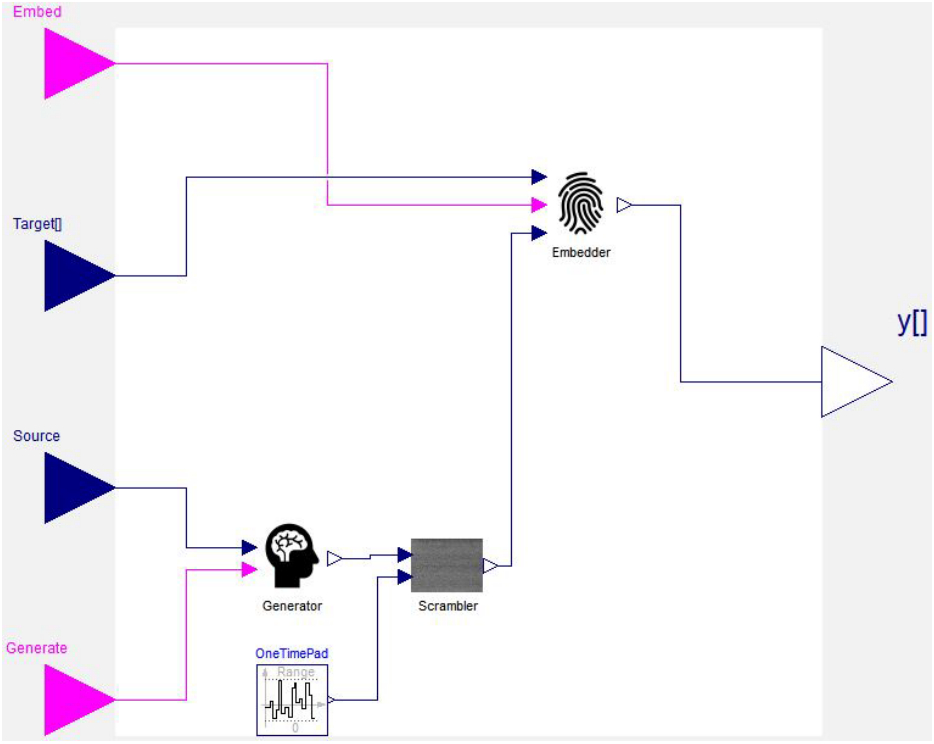

Figure 21: Dymola C[2] modules for message generation, obfuscation and embedding

The generator consists of a dimensionality reduction algorithm, principal component analysis (PCA), to extract only the dominant components of the source sensor in real-time. While the procedure may be generalized to multiple sensors, in this instantiation, information about the recent temporal evolution of the reactor power sensor (see Table 2) is extracted and sent to the scrambler. The dimensionality reduction algorithm may be implemented dynamically or done offline to construct the observable and non-observable space depending on the available resources.

Typically, incoming information is patterned since it describes the temporal evolution of the source sensor. However, since the non-observable space of the target sensors are typically noisy, this information must be obfuscated to remove patterns and transformed to fit the noise distribution of the target sensors to remain covert. The one-time-pad cryptography transforms the input information into a random version using a randomly generated pad. It is information-theoretically secure as long as the pad is random, not reused and only shared with the receiving party, and the randomized version may be inverted to the input information using the pad. The scrambler uses an algorithm similar to the one-time-pad with a source of randomness to render the incoming information patternless and sends it to the generator. Additionally, the one-time-pad

constantly changes with time, so the same incoming information being embedded at two different time instances will have different scrambled outputs. This is done so that an AI/ML tool cannot decipher the underlying algorithm and will be unable to find any association between the inputs and outputs.

Once the information is scrambled, the embedder has its own dimensionality reduction algorithm to extract the observable and non-observable space of the sensors. In addition to extracting the non-observable space, the embedder also analyzes the target distribution of the carrier variables prior to embedding. These are independent, identically distributed samples of white-Gaussian noise and thus, the incoming scrambled information is scaled to fit the distribution of the target carrier variables. In the present work, since the ten sensors (in Table 2) are correlated and reducible to fewer degrees of freedom, the remaining degrees of freedom compose the non-observable space for embedding the temporal information of the power sensor from the generator module. Note that the non-observable space may be computed online or offline depending on resource constraints similar to the generator module.

With regards to intrusion detection and data recovery, the process is simply reversed since all transformations are invertible. The scrambled information may be extracted from the sensors since the non-observable space is known. If the scrambled information does not match the actual output of the scrambler, the data has been compromised and an alarm is indicated. In this case, the expected output is unscrambled using the one-time-pad algorithm to extract the original information. Since this information corresponds to the dominant behavior of the temporal data, the power sensor data may be reconstructed, allowing the system to continue to operate with the reconstructed data instead of the incoming tampered data.

There are several extensions which may serve as additional layers of security and recovery. For instance, randomizing the “Generate” and “Embed” pulses serves as another layer of security for the SMR, since it is impossible to predict when the pulse may trigger. Along the same lines, the chosen courier variable in the embedder module may be randomized, or due to the vast dimensionality of the non-observable space, the same information may be embedded along multiple courier variables using different one-time-pads as backup. The link between the source and target sensors may also be randomized to ensure an adversary intimate with this specific implementation neutralized.

4.2.2 Numerical Experiments and Results

In this section, the automated C^2 modules are implemented and validated via an IRIS SMR design developed using the TRANSFORM library in Dymola. The zero-impact and zero-observability criteria of the C^2 paradigm are first validated followed by a demonstration of the intrusion detection and data recovery capabilities. The reactor is simulated under various operational conditions by adjusting the target power between 50 and 100% of the plant capacity. A typical operational mode involves steady-state operation, followed by two linear ramps culminating in a parabolic load-following behavior as shown in Figure 3.

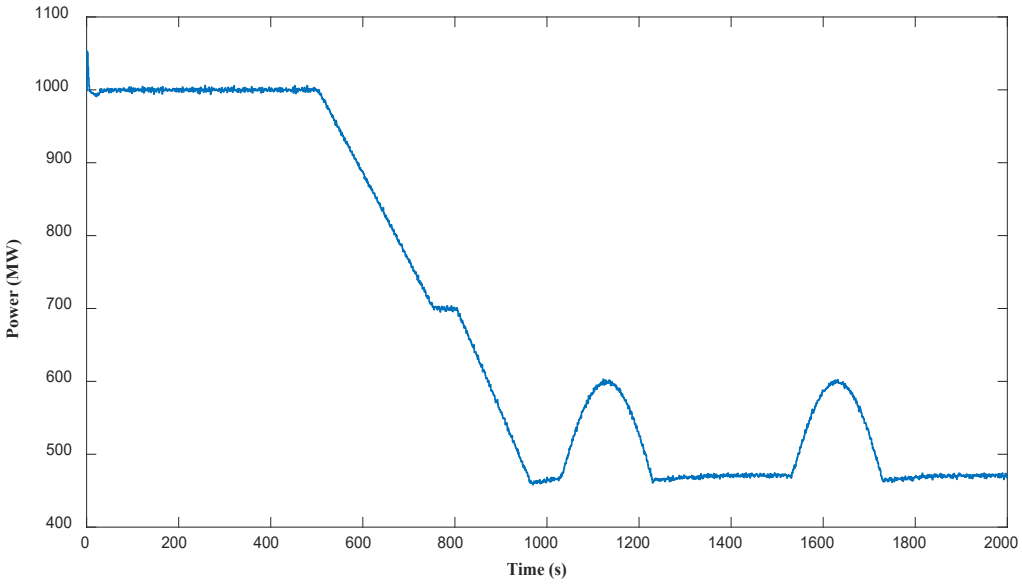

Figure 22: Representative operation of IRIS SMR.

The C^2 embedding module procures information about the temporal evolution of the power sensor and embeds it across all the 10 sensors outlined in Table 2. The process is done every 10 seconds via the "Generate" and "Embed" pulses and using Eqs. 4-6. To validate the zero-impact criterion, the controller cost is computed using the sensors and actuator inputs over 10 different simulations. Of these, the first five simulations represent various operational modes as seen in Figure 4, while the latter five simulations validate the zero-impact criterion over different instantiations of noise (different random numbers) for a given mode to further showcase that the change in operational cost due to C^2 is within the noise variance of the controller cost as seen in

Figure 5. As such, a classifier based on the controller cost alone is expected to be unable to distinguish between the two processes.

The controller cost, borrowing from control theory and described in further detail in previous work, is a measure of the effort expended to change a process from an initial state to a desired state. It has arbitrary units and is a function of the state-space, the control inputs, and the error between the desired state and the current state, with cost minimization being the goal of most control system designs. In this set of numerical experiments, however, the PI controller parameters were already provided so no optimization was done. The cost was simply computed as the sum of the states and the control inputs across all time-steps with appropriate scaling. Note that this is arbitrary and the following analysis follows without loss of generality for different cost functions since the effect of C^2 on the operational data is within the noise level. Effectively, regardless of specific cost function, the variations in controller cost due to the C^2 modules are within the variations expected due to randomness of noise.

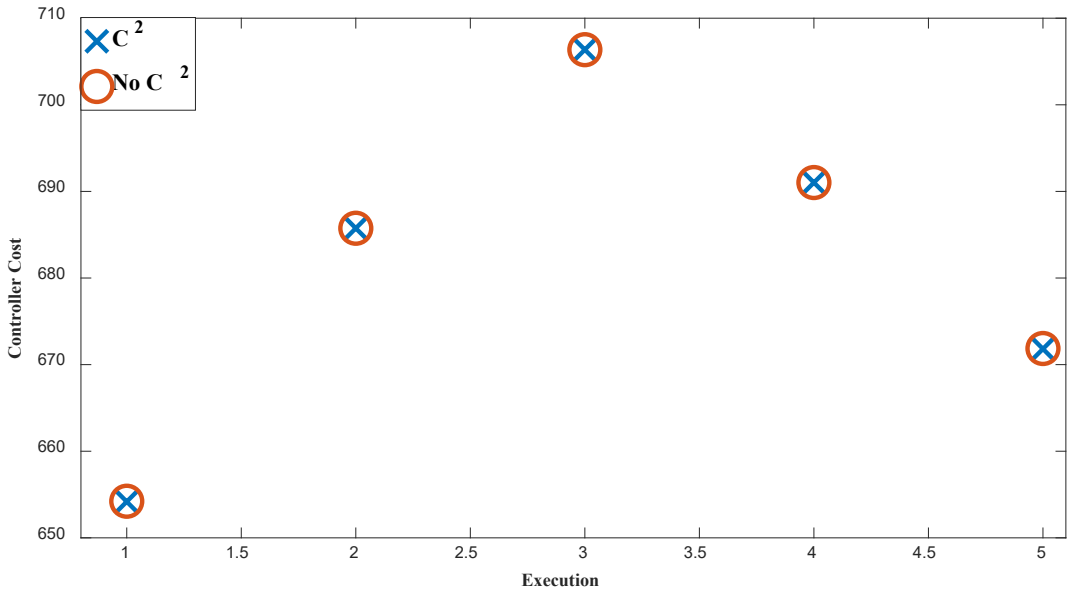

Figure 23: Controller cost over five runs with different operational modes.

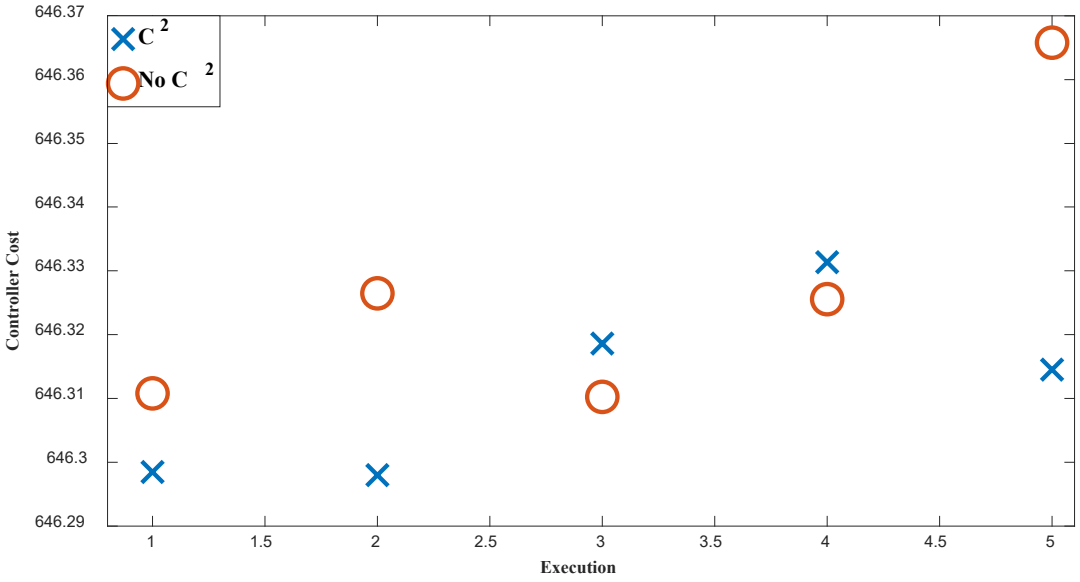

Figure 24: Controller cost over five runs with different noise instantiations for a given mode.

The zero-observation criterion is validated visually by viewing the analyzing along the non-observable space and statistically validated using the Kolmogorov-Smirnov test at a significance level of 0.01. By design, the C^2 paradigm does not affect the dominant space as seen in Eq. 6. However, the statistics of the coefficients along the non-observable space vector in Eq. 5 must be preserved. To this end, the coefficients of the vector are computed and plotted in Figure 6 before and after embedding at the time instances where the embedding occurs. It is observed that the embedded coefficients are within the existing noise-level, thus validating non-observability. To summarize, Figures 4-6 validate the claim that any change to the operational data from the C^2 embedding (Eqs. 4-7) is within the noise of the data arising out of measurements. Further validation of the C^2 paradigm using the underlying probability distribution of the reactor states, outputs, inputs etc. was done in previous work via a generative adversarial net with two competing long short-term memory networks [41].

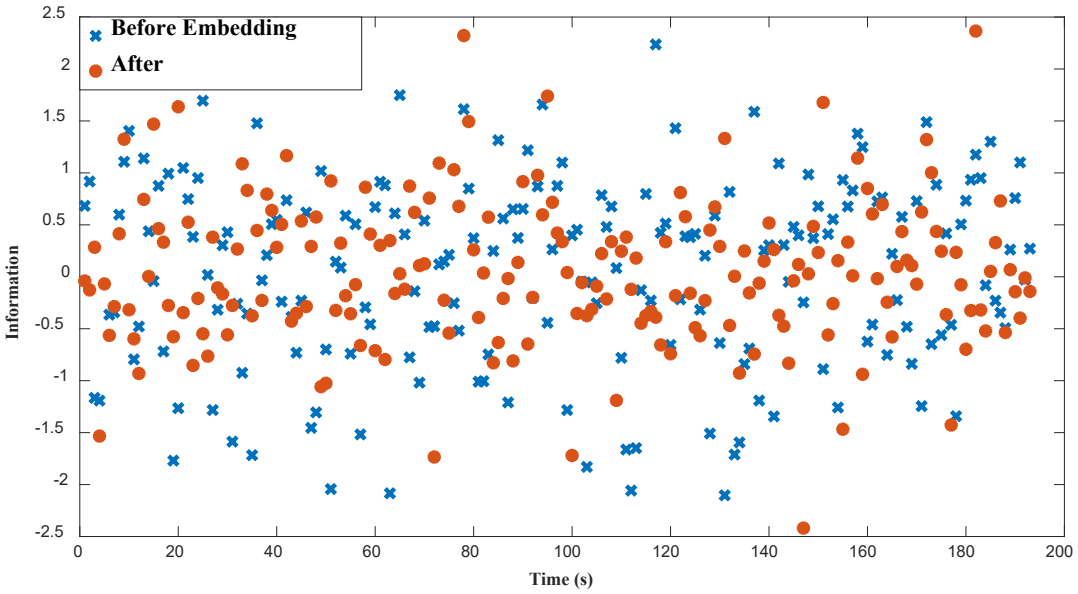

Figure 25: Coefficients along non-observable space.

The primary focus of this work, however, is on the intrusion detection and more critically, data recovery capabilities of the C^2 modules under compromised environments. To accomplish this, first, we simulate a replay attack by falsifying all 10 sensors using past data. In this work, we consider a reactor operating under steady-state conditions with all its sensors falsified after 500 seconds with the ramp data from an earlier simulation. Although all 10 sensors are falsified, only the four sensors that affect the PI control algorithm are shown for brevity in Figure 7. While falsification of a subset of sensors may be detected via redundant or additional sensors, since correlations may not be preserved, falsification of all sensors preserves the expected physical correlations, and is thus, expected to bypass passive algorithms that rely on the physics of the data or a digital twin, since the false data comes from an earlier "genuine" scenario. Additionally, residual-based detectors such as χ^2-detectors when combined with passive methods are also bypassed since the noise statistics are preserved in a replay attack.

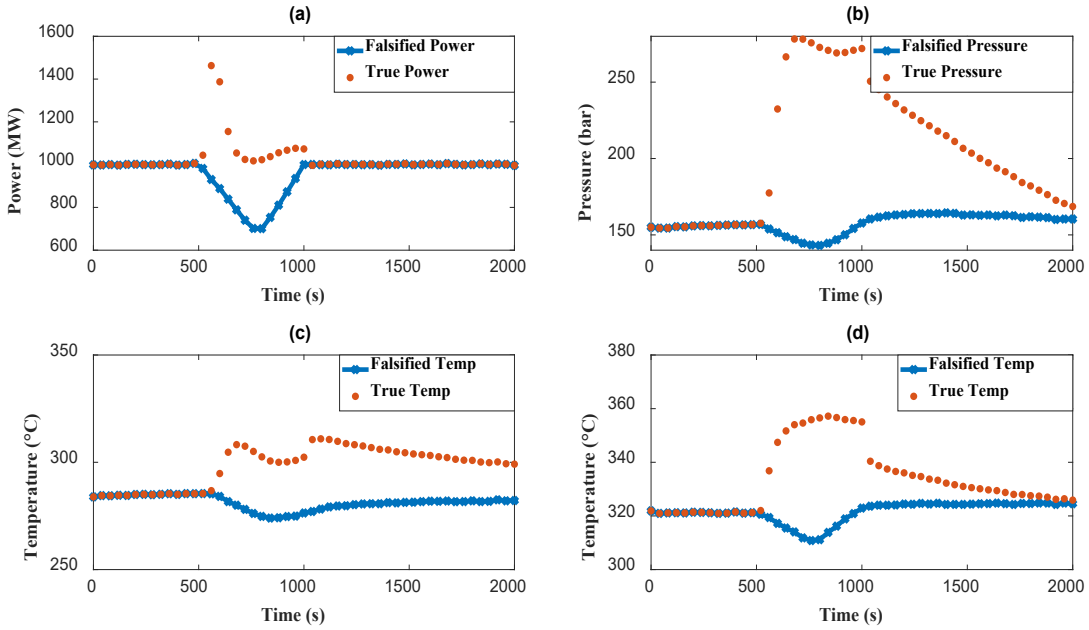

Figure 26: Effect of replay attack; left to right, top to bottom – 7a) Reactor Power, 7b) Reactor Pressure, 7c) Reactor inlet temperature, 7d) Reactor outlet temperature.

As seen in the four subplots of Figure 7, the replay attack is capable of driving the reactor to unsafe states since the PI controller attempts to return the reactor to steady-state power based on the falsified input. Specifically, the reactor power is driven to 1500 MW, well above its maximum output of 1000 MW, as seen in Figure 8a. Note that additional falsification of the actuator command display to the operator is possible to further obfuscate the attack. Regardless, a replay attack is shown to be highly disruptive to operations; the reactor ramps up to 150% of its capacity as seen in Figure 7a, while appearing to be a regular load-following transient, with a corresponding rise in the pressure (7b), inlet (7c) and outlet temperature (7d). The attack may also be rendered subtle via minor falsifications of the reactor state over a time causing the reactor to approach an unsafe operational regime in a subtle manner.

However, with the addition of the C^2 module, it is observed that the noise analysis across all sensors is expected to carry information of the temporal evolution of the reactor power. Additionally, due to the one-time-pad, the same information is expected to be obfuscated differently at different time-steps, i.e., as seen in Eq. 5, even if the extracted information z_i is identical across time, the one-time-pad obfuscation function $r_i(\cdot)$ changes and thus the output of the scrambler module is different. In other words, this implies that a replay of past sensor data is

insufficient to bypass the detector since the pad required for unscrambling is different. Thus, the proposed approach serves as an intrusion detection mechanism as seen in Figure 8 since the recovered information using the non-observable space operator does not match the expected result at the first instant of falsification.

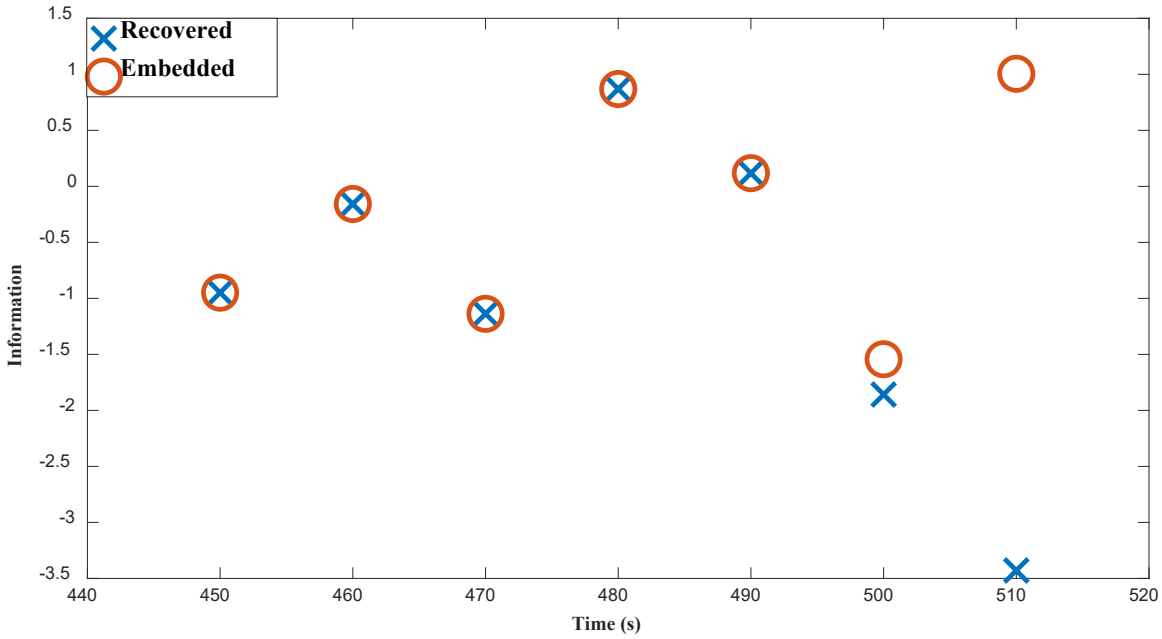

Figure 27: Instantaneous detection of replay attack using embedded information.

The data recovery functionality is demonstrated by the recovery of the embedded information by reversing the one-time-pad obfuscation as seen in Figure 9. Using the recovered information, the original time-series may be reconstructed using Eq. 4 since the information was extracted from the temporal evolution of the power series. For verification, the embedded and recovered information are plotted against each other in Figure 10 assuming that the detection algorithm detects the replay attack as shown above and switches to recovering the actual data and continuing operation.

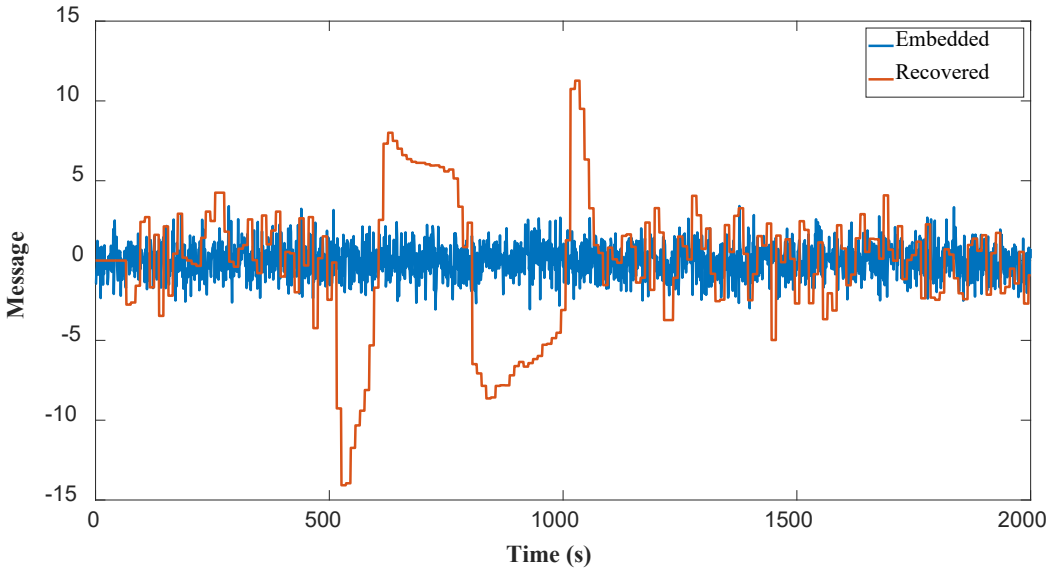

Figure 28: Recovery of embedded information using one-time-pad.

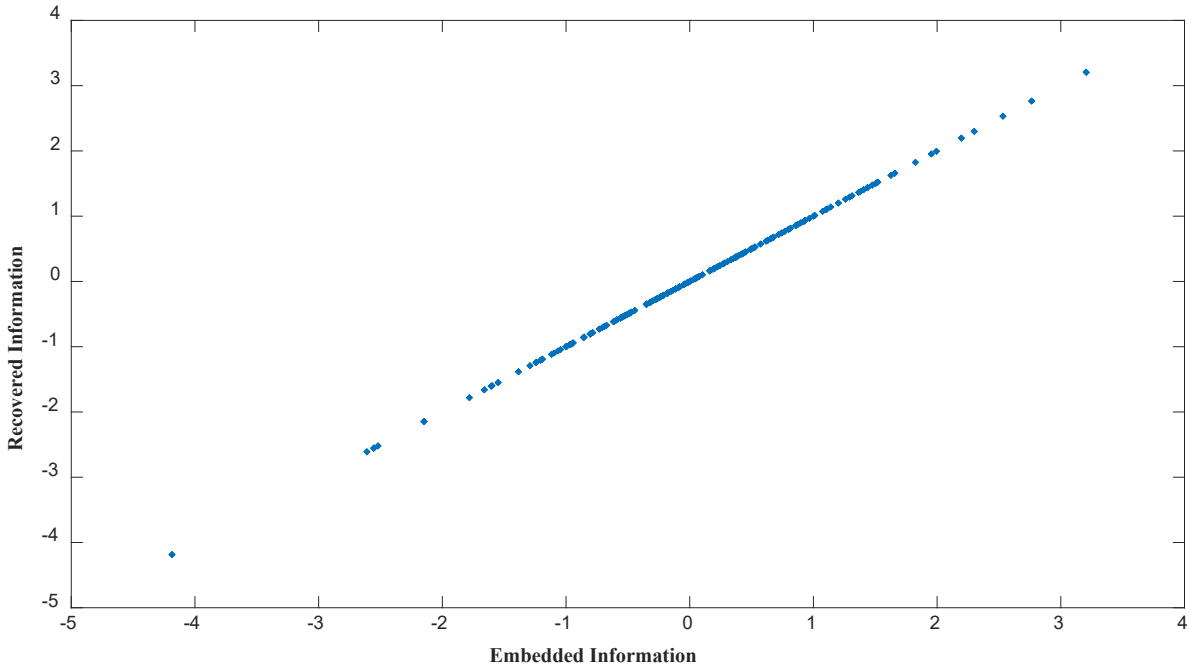

Figure 29: Complete recovery of embedded information.

In the above experiments, the C^2 modules have been implemented and validated in an IRIS SMR simulated using Dymola. Specifically, the zero-impact and zero-observability of C^2 were first validated under normal conditions and various modes of steady-state and transient operations. It is observed that the controller cost varies within the noise-level with C^2 and the noise level along the vectors in the non-observable is also preserved. Then, a replay attack is simulated on the Dymola module using previous simulation data to represent intrusion by a skilled adversary, such as an insider, and it is detected instantly using the automated C^2 modules. Furthermore, the data recovery capabilities of C^2 is validated by extracting the embedded information from the non-observable space, reversing the one-time-pad operation, and reconstructing the original time-series. With regards to fission batteries and microreactors, the above modules have been developed in a stand-alone plug-and-play manner and may easily be integrated into the software and/or the hardware design of the instrumentation. After an initial training period, the generator and embedder module perform dimensionality reduction on the input sensor data, identify the observable and non-observable space, and embed the information for recovery during contingencies such as cyberattacks. As demonstrated, the method also readily adapts to significant changes in operational modes, which is not expected of fission batteries that typically operate at steady-state. In this case, the training and subspace identification may be done offline and hardcoded into the fission batteries' instrumentation, with only the embedding and recovery processes carried out online during deployment.

4.3 Discussion

One of the key requirements of unattended operation in fission batteries is the ability to automatically verify the integrity of sensor data and undertake data recovery actions in the face of cyberthreats with minimal human intervention. The key contribution of the present work, in this regard, is the implementation of covert cognizance (C^2) to endow microreactors with this ability by inducing awareness of its own operational history and permitting data recovery capabilities if the system is compromised. The IRIS SMR design in Dymola, chosen as a representative module, was simulated using the TRANSFORM package and its sensors were perturbed in accordance with the zero impact and zero observability criteria set by the C^2 paradigm. The C^2 modules are composed of a generator module that applies dimensionality reduction algorithms to extract core information from a source sensor, a scrambler module that obfuscates and transforms the

information into noise, and an embedder module that embeds the noise along the non-observable space of the other sensors. In the event that the source sensor is compromised due to a cyberattack, an intrusion detection and data recovery algorithm based on the C^2 modules can deterministically pinpoint the attacked sensor, recover the lost information using the non-observable space of the other sensors, and ensure normal operation of the system while a response team deals with the cyberattack. The implementation is then validated by simulating a replay attack by a knowledgeable/skilled adversary that falsifies all existing sensors by simply replaying data from previous operational conditions. While model-based detection schemes can detect falsification attacks that do not preserve physical correlations between the various sensors [31], the simulated cyberattack evades detection by simply replaying past data across all sensors, thus preserving physical correlations and representing the highest level of access an insider/advanced persistent threat actor may have to the control system.

Without the C^2 modules, it is demonstrated that the reactor could be driven to an unsafe state well above its operational regime while appearing to adjust to a decrease in power based on the falsified sensors. This could have disastrous implications in a real scenario, potentially leading to shutdown of the reactor if the emergency cooling systems are activated and even a meltdown scenario if the emergency systems are linked to the falsified sensor measurements. When augmented with the C^2 modules, the intrusion is detected deterministically, and the actual power sensor measurements are recovered using the embedded information from the other sensors. The present work explores an application of the C^2 technology to enabling automated data recovery and unattended operation in fission batteries in the face of cyberattacks. The novelty lies in the ability to recover data in a compromised environment by working within the constraints of the physical systems and without the introduction of additional variables, redundant sensors, backup devices etc. This contrasts with existing data recovery tools that rely on restore points, vault-like storage of raw data, multiple copies of the data, etc., to restore operation after a cyberattack and whose security hinges upon the lack of access to these vaults. The designed C^2 modules exploit the inherent redundancy of dynamical systems to store information within the existing variables, do not mandate system downtime, and provide an avenue for automated recovery. While the present work is intended as a prototypical rendition, further work is necessary to develop the architecture, i.e., the placement of the modules within a system, secure and covert communication channels for verification and validation, setting up the intrusion detection and data recovery

channels etc. Future work is expected to focus on the integration of C^2 with established testbeds for microreactor applications such as the MARVEL testbed proposed for proof-of-concept demonstration.

The present work showcases a hypothetical digital implementation in the control system of a small modular reactor. However, the developed C^2 modules are standalone and may be easily integrated into the software of a digital twin and/or embedded into the hardware instrumentation of a microreactor/fission battery. As part of future work, plug-and-play hardware modules with the C^2 modules embedded shall be developed and deployed onto real testbeds such as MARVEL to automatically induce self-awareness in these systems and augment them with the ability to recover data and operate autonomously during a cyberattack. As an intermediary step, hardware-in-loop simulations supported by Dymola may be used to virtualize a microreactor environment and test the hardware C^2 modules.

5. CASE STUDY: DATA MASKING

This chapter extends the C2 paradigm to data masking applications through the novel deceptive infusion of data (DIOD) methodology and borrows from content published in the Transactions of the American Nuclear Society [37], and the journals, Nuclear Science and Engineering [140] and Nuclear Technology [141]. Recent decades have seen a growing demand of AI/ML data analytical services that can help businesses optimize their resource allocation strategies and maximize revenue through data-driven insights. The global big data and business analytics market was valued at nearly $200 bn in 2020, with projected estimates of $684 bn by 2030 [142] with various subsections focusing on predictive analytics, prescriptive analytics, descriptive analytics, customer analytics etc. In industrial systems, there is a strong need to integrate AI/ML insights with process data for applications such as condition monitoring, vulnerability analysis, autonomous control etc. To realize this goal, it is critical that AI/ML services are granted access to sensitive process data, often to the chagrin of owners of proprietary systems that are reluctant to share details of their systems for fear of loss of competitive edge or discovery of vulnerabilities that may be leveraged against them. The present work identifies a need for a data masking technique for industrial data that preserves the necessary inferential properties for AI/ML services while obfuscating sensitive information that may provide clues to the identity of the proprietary system. To this end, the deceptive infusion of data (DIOD) data masking paradigm is proposed to enhance collaboration among owners of critical infrastructure and third-party AI/ML services to leverage the analytical capabilities of AI/ML. Specifically, it is recognized that industrial data may be decomposed into the so-called fundamental and inference metadata describing the system identity and process parameters respectively. The fundamental metadata may be obfuscated to achieve data masking goals, while the inference metadata is preserved using the idea of mutual information to ensure that the data is relevant for AI/ML applications.

This work may be considered as an offshoot of the C2 decomposition work done in identifying an observable and non-observable space for industrial systems and is motivated by the physical meaning behind the active DOFs and their components. This chapter is organized as follows: section 4.1 describes the background of existing data masking techniques and demonstrates the need for DIOD, section 4.2 provides a mathematical framework for the

implementation, section 4.3 provides an application for DIOD to create AI/ML benchmark datasets, and lastly, section 4.4 presents DIOD as a solution to reverse-engineering.

5.1 Background

Business intelligence [143], [144] is a technology-driven process in enterprises seeks to merge data analytics with business information in order to maximize shareholder value. Its goal is to provide a concise interpretation of data and transform it into actionable insights for business strategy through data mining, process analysis, performance benchmarking etc. Such insights often fall under the broad umbrella of business analytics, the area of analytics that focuses on statistics, optimization, and prediction for enterprises. Owners of proprietary systems such as critical infrastructure often seek the additional insight provided by AI/ML services to remain competitive in the market. This is especially relevant in the nuclear industry that competes against relatively less sensitive renewable technologies such as solar, wind, etc.

However, vendors of AI/ML services are often third-parties that may not respect the need for data privacy as required by owners of proprietary systems, who are often reluctant to share process data for fear of data leaks and misuse. For example, an AI/ML service focusing on detecting signs of equipment degradation from sensor data may discover additional loopholes and security vulnerabilities by reverse engineering the data to find information related to the design of the system. Unscrupulous services may subsequently sell this information to the highest bidder that can carry out cyberattacks targeting critical vulnerabilities in the system. The demand for such information remains high; in recent years, hacker groups such as REvil have recruited affiliates to conduct multiple high-profile ransomware attacks against tech giants such as Apple, academy trusts, defense contractors, meat processing plants etc., earning millions in the process. Recent research on the effects of data leaks by Accenture [145], [146] indicates a 10% drop in revenue for six months following public exposure of private information, necessitating the need for data masking techniques to protect sensitive information in the event of a breach to ensure that the information cannot be traced to their owners. The following subsections describe various reverse-engineering tools that may be used to extract sensitive information, the extant data masking solutions to combat these techniques and their shortcomings with regards to industrial data, and lastly, the DIOD paradigm that extends the application of data masking from data warehouses to industrial data.

5.1.1 Reverse-Engineering in Industry

AI/ML services have been at the forefront of research in recent years. The advent of smart technology and the widespread incorporation of AI/ML techniques into critical infrastructure has led to the massive benefits such as improved operational efficiency, predictive maintenance etc. However, such tools have also been leveraged by malicious agents that exploit their pattern-detection capabilities to reverse-engineer critical systems and learn proprietary information. Consequently, stakeholders often face the conundrum of leveraging the benefits of AI/ML without leaking potentially proprietary information to third-party agents. Additionally, the core dynamics of critical industrial systems, especially modules such as powerplants, and steam generators, is widely disseminated in academic and public circles. The release of industrial data is thus heavily scrutinized, and the data is often sanitized prior to release to conceal critical physical parameters. The extraction of such parameters is often portrayed as an inverse problem, i.e., given a set of responses, and a basic model of the system, is it possible to estimate the concealed parameters? The following section provides a comprehensive literature review on state-of-the-art AI/ML tools that have reverse-engineering capabilities in the context of industrial systems.

Historical approaches to solving inverse problems were often knowledge-driven and required accurate analytical models of the physical system. Regularization techniques such as Tikhonov regularization were employed to account for uncertainties in data [147]. However, if the inverse problem is ill-posed, i.e., the number of response variables is much smaller than the number of parameters, there may not be a unique solution. Furthermore, obtaining a reasonable solution is often computationally infeasible with analytical methods, requiring data-driven approaches. Data-driven methods [148] such as neural networks have been utilized to solve the problem of parameter estimation for over 30 years and circumvent the problem by learning by example, i.e. the network parameters are trained to fit the data without any model. Some of the earliest applications of neural networks involve the estimation of soil parameters such as water table elevation and concentration [149]; however, the lack of adequate storage and data processing capabilities were often a hindrance to the implementation of deep models. In recent years, with the advent of big data analytics, DNNs have seen a re-emergence of such techniques to solve inverse problems. For example, GANs have been utilized in 3D image reconstruction using CT, PAT and MRI images [150]–[152]. In some cases, however, there may be inadequate data to sufficiently train the neural network due to resource constraints. To this end, recent research hybridizes both knowledge-driven

and data-driven approaches to develop physical-analytical models and mitigate some of the shortcomings of both models. The so-called physics-informed neural networks [153], [154], while still in their infancy, have shown promise in parameter estimation in fluid dynamics by identifying unknown parameters using the Navier-Stokes equations and flow velocity measurements. In this approach, the physical laws of continuity and conservation principles are directly encoded in the loss and activation functions of the neural networks, which must be minimized. For example, conservation of momentum may be encoded as the mean squared difference of the momentum between the two consecutive nodes. Additionally, since the networks learn by gradients, time and space derivatives may also be directly implemented in the network. The parameter estimation work has also been extended to biological systems [9], [155].

ROM is another popular reverse-engineering tool where low-order models are constructed to approximate complex dynamical systems [38], [45], [46]. As mentioned in the motivation behind the C2 paradigm [64], such systems are often inherently reducible, with their dominant behavior described only using a few active DOFs. Research has shown that even if the true model is concealed, neural networks based on the input and output data may be constructed to learn the underlying model. The challenge is greatly simplified for an adversary if the system dynamics are well-known, as is the case with nuclear powerplants whose parameters may be estimated with simple point-kinetics models that can be found in any introductory textbook on nuclear engineering [39]. With sufficient data, a capable adversary may bridge the gap in knowledge and successfully glean information about the underlying parameters that must be kept confidential. In fact, artificial neural networks have been used to reverse-engineer gene networks in biological systems to identify complex higher-order dependencies between expression patterns [156]. Although such dependencies are difficult to capture using brute-force approaches due to the prohibitively large search space, neural networks have been shown to circumvent this by searching within the neighborhood of each gene.

While there is a lot of empirical evidence supporting the use of deep learning to solving inverse problems, the limitations of such approaches are largely unknown due to the relative obscurity of the inner workings of deep networks. Over the past decade, attempts have been made to disrupt the learning process of deep networks via minor perturbations in the data that maximize their training loss and induce a misclassification [59]. Aptly called adversarial attacks, these attacks appear like noisy perturbations to the human eye but have a disproportionate impact on the

performance of deep neural networks. However, recent research [157] appears to indicate that due to the inherent nature of inverse problems, a few deep networks have been shown to be robust to uncertainties arising out of noise as well as adversarial attacks.

Intelligent perturbations of dynamical systems [158] have been proposed as a candidate for automated reverse-engineering of complex dynamical systems. Candidate models may be suggested based on domain-knowledge or observing common mathematical relationships among the variables and their derivatives, from which an optimal model is chosen via perturbations. These models may uncover fundamental laws of nature such as the Hamiltonian or Newton's laws, through symbolic manipulation while others such as PDE-FIND [51] utilize sparse regression techniques to uncover partial differential equations. They have also been augmented by AI/ML to decipher underlying physical laws with a relatively small amount of training data. Similar capabilities have been observed in the context of 3D-printed materials, where the microstructure of such materials and the tool path of the 3D-printer was inferred from CT scan measurements using a recurrent neural network [159].

In addition to industrial data, recent research in computer science indicates that even executable code may be reverse-engineered through the use of AI/ML techniques [160]. This is especially beneficial to an adversary in the event that the data from a simulation is inadequate for parameter estimation, and one is unable to access the source code due to proprietary restrictions. In such cases, it is possible to recognize certain functions during execution via observation of the assembly code using CNNs. While the research is in its infancy, this presents a new challenge to industrial stakeholders as reverse-engineering in the context of binary analysis is typically considered prohibitive.

Reverse-engineering and data leaks are expected to cost companies billions of dollars in annual revenue. Consequently, nearly 40% of firms redact proprietary information prior to going public due to fears of losing competitive edge [161]. Moreover, research in the financial sector has shown that these firms often outperform their disclosing counterparts and exhibit greater profitability despite initial underpricing and investor concern due to the lack of information. In the context of industrial data, owners of proprietary systems often face the dilemma of sharing data at the risk of leakage to rival firms, necessitating the use of data masking techniques as outlined in the following section.

5.1.2 Data Masking

The need for data masking has historically followed the adoption of data warehouses by enterprises to store massive amounts of information typically related to business logistics. With the rise of "big data" and the data hungry nature of AI/ML tools, enterprises often rely on these warehouses to remain competitive, making them an attractive target for hackers and malicious insiders. Consequently, the transmitted data often undergoes a comprehensive sanitizing process using data masking techniques such as substitution, shuffling, differential privacy etc. [162]–[166] that seek to protect sensitive information pertaining to the proprietary system. In substitution, sensitive information may be obfuscated by substituting it with seemingly irrelevant information while the relationship between the sensitive and substituted information is only known to the implementer of the data masking paradigm. Associations between data fields may be permanently altered by shuffling records to ensure that even if a subset of information is leaked, it cannot be linked further to other data fields to fully reconstruct the original record. For example, if an application involves analyzing the gender distribution of employees in an organization, the names may be substituted with generic names, and the age may be shuffled to prevent further inference being made on the age, SSN etc. of the employees. These methods may be considered as static since they permanently alter records, requiring the creation of an additional database (a copy) with the original sensitive information. Dynamic methods, on the other hand, only mask information on the receiver end and may work in real time with the caveat that they may be susceptible to corruption and data loss since the original record is transmitted. Common examples include the masking of most characters in a retrieved email ID, the initial digits of SSN etc.

Differential privacy is an emergent data masking technique that statistically perturbs the data to prevent adversaries from piecing together information to reconstruct the original record. Relying on the fundamental law of information recovery – "overly accurate answers to too many questions will destroy privacy in a spectacular way" – the method introduces uncertainties to protect the privacy of individuals in a group. The statistical distortion is expected to have a minimal impact on the group statistics/inferential properties required by AI/ML services. A common example of differential privacy is that of randomized survey techniques. Survey takers are encouraged to take sensitive survey questions (vices, habits etc.) by anonymizing their responses through a discreet coin-toss. Depending on the face of the coin, the survey taker is asked to either take the survey truthfully or select a pre-determined response. Based on an individual response, it

is impossible to deduce whether it was truly indicative of the survey taker's habits better than random. Nevertheless, population statistics may be deduced using probability theory with a large enough sample size, thus protecting the individual's privacy while providing relevant information for the survey. However, for large datasets with multiple uses and statistical parameters to be gleaned, the effects of the individual perturbations may have a significant effect on the utility of the entire dataset, rendering it ill-suited for industrial applications.

Along the same line, privacy-preserving computation techniques such as fully homomorphic encryption [167] have been popularized to enable mathematical computations on datasets without decrypting individual records. For instance, consider the simple computation of $2 + 3 = 5$. With homomorphic encryption, an addition operator is designed such that one might add the encrypted versions of 2 and 3 to obtain an encrypted 5, which may be subsequently decrypted for the desired application. Such methods, however, are in its infancy, and are currently being explored for applications in the healthcare industry, where patient records may be manipulated without identifying the individual patient. Additionally, the high overhead costs of encryption remain a significant obstacle to commercial feasibility, especially for AI/ML applications that require the mathematical manipulation of vast amounts of data.

The above data masking schemes, while suitable for data warehouses, fail to fulfill the needs of industrial data. This is because it is infeasible to simply mask or scale time-series data from the process variables without preserving existing correlations. Here, the goal is to preserve the inferential properties of the data while preventing reverse-engineering techniques from associating the data to its source – the proprietary system. For example, the optimization of parameters for a turbine design may involve certain correlations among the blade speed and length based on the underlying physics. These correlations must be respected while masking to ensure similar statistical inference on the masked data and the original data. Due to the high frequency of data collection, it is often only feasible to strip datasets of some critical process variables and encrypt them prior to transmission with no measures in place to protect the data from unscrupulous third-party AI/ML services once decrypted. Additionally, the stripping of critical process variables or parameters may affect the usability of the data and degrade the performance of the AI/ML algorithm, proving to be counterproductive to the goals of data sharing. To protect the data, the sharing of sensitive information is consequently surrounded by bureaucratic red tape with multiple disclosure forms and agreements to ensure that the receiving party may be held accountable in the

event of a breach. However, this may be impractical and/or inadequate and does not prevent the damage from a leak which may be devastating to the proprietary owner and typically not limited to economic consequences. While recent years have seen the implementation of data privacy laws such as the California Consumer Privacy Act [168] and the General Data Protection Regulation [169], legislation is still in its infancy and often suffers from jurisdictional concerns. Additionally, such laws often make it difficult on the proprietor end to share sensitive data since they are held responsible for the handling and dissemination of the data. This motivates the following question: Is it feasible to develop a data masking methodology for industrial data that preserves the inferential properties of the data while masking system identity to remove sensitivity concerns?

5.1.3 Deceptive Infusion of Data (DIOD)

The proposed DIOD paradigm [37] addresses the above issues through a novel time-and-space scalable data masking methodology that relies on the decomposition of industrial data into its constituent metadata, namely, fundamental and inference metadata. The fundamental metadata are tied to the identity of the system such as the underlying physical model, the associated differential equations, etc. The inference metadata, on the other hand, contain information on the operational conditions such as process parameters, boundary conditions, initial conditions etc. that are relevant for various AI/ML tasks like classification, regression, and optimization. Through a novel mathematical algorithm, the fundamental metadata of a proprietary dataset is replaced by that of a generic system, thus changing the identity of the parent system itself. The DIOD dataset is subsequently generated through the fusion of the fundamental metadata of the generic system and the inference metadata of the proprietary system. This is done through the use of so-called "deception kernels" developed using a library of concealment operators that may be developed using the ROM techniques outlined in the C2 paradigm. The DIOD paradigm contrasts with existing methods in that it does not affect the inferential properties of the data as done by substitution, shuffling etc. and only requires an upfront cost to develop the concealment library as is the case with building ROMs. Thereafter, it provides a scalable solution to data masking since generated datasets may simply be obfuscated using the concealment kernel at a substantially reduced computational cost compared to encryption. Additionally, unlike encryption, reverse engineering of the DIOD dataset does not provide any clues to the identity of the proprietary

system since that link is permanently altered by the deception kernel that is only known to the owner of the proprietary data.

Furthermore, attempts to reverse the deception process is mathematically infeasible due to the vast number of candidate proprietary systems and possible metadata. This may be better illustrated through an analogy in the control community regarding state-space models and transfer functions. The transfer function for a given state-space model is unique while the inverse does not hold true. A given transfer function may be the result of any one of infinite possible state-space models, rendering it impossible to guess the original system. Analogously, unless the exact deception kernel is known, it is not possible to guess the fundamental metadata of the proprietary system given the fundamental metadata of the generic system that may be extracted from the DIOD dataset through reverse-engineering. A few interesting questions that arise from this discussion are: Could the inference metadata perhaps provide clues to narrow down the parent system? If so, could it be obfuscated without affecting the inferential properties of the dataset?

Certain systems may have inference metadata possessing correlations or process parameters that may be unique to the system or can significantly narrow down the possible candidates given their proprietary nature. This is a significant challenge for the DIOD work since manipulation of the inference metadata may render the data unusable for target applications. To this end, the concept of mutual information [170] is leveraged to provide additional layers of obfuscation that can mask the ties between the inference metadata and the proprietary system. Mutual information is a statistical entropy-based measure that captures all the linear and nonlinear dependencies between two variables. In the case of classification, the variables may be the classifier label and the extracted process parameters. The problem then reduces to finding the subset of parameters that carry relevant information while discarding the remaining. A key advantage of mutual information is its invariance to invertible mathematical transformations and the addition of extraneous information. In fact, AI/ML tools such as kernel SVMs [55] exploit this property to project low-dimensional data onto a high-dimensional space and draw a decision boundary (hyperplane, hypersphere etc.). This provides an avenue for masking the inference metadata further to closely represent the properties of the generic system. For instance, suppose the proprietary system is a nuclear reactor with reactivity being the relevant inference metadata and the generic system is a spring-mass system with a certain spring constant. The reactivity, being a real number, cannot be directly used as a spring constant since the latter is a strictly positive

quantity. Here, invertible transformations such as scaling and shifting may be used to alleviate this issue without loss of information for the target AI/ML classification task, as explored further in the following section on the mathematical framework.

The DIOD data masking paradigm has several applications. First, it is recognized that the same inference metadata may be fused with the fundamental metadata of multiple generic systems using various deception kernels from the pre-developed library of concealment operators to create a benchmark DIOD dataset for AI/ML applications. Theoretically, since the fundamental metadata is irrelevant to the AI/ML task, an ideal algorithm must be invariant to the fundamental metadata and achieve similar performance on all the datasets. This application is explored in section 4.3 where the inference metadata from a reactor is directly fused with the fundamental metadata of a direct current permanent magnet (DCPM) and validated using various AI/ML tools such as supervised algorithms, unsupervised clustering algorithms, singular value decomposition, and response correlations.

Secondly, recent years have overseen the use of open-source environments and datasets to create a collaborative environment among various AI/ML researchers. Through competitions hosted on websites such as Kaggle, enthusiasts are encouraged through reward-based systems to develop novel architectures and data mining tools to perform unsupervised and supervised AI/ML tasks. However, owners of critical infrastructure are unable to leverage the benefits of open-source collaboration due to the sensitivity of their data and the associated risks as mentioned in section 4.1. Using the DIOD paradigm, however, this sensitivity factor is eliminated since reverse-engineering of the data only leads to the identity of a generic system, thus providing an avenue for owners of critical infrastructure to participate in such collaborative environments. This application is explored in section 4.4 where various constraints based on the mutual information metric are developed to achieve multiple layers of masking on the inference metadata as well in addition to the fundamental metadata obfuscation in the previous application. The following section provides a mathematical framework for the DIOD paradigm using the concepts of fundamental metadata, inference metadata, deception kernel, and mutual information.

5.2 Mathematical Framework

The decomposition of industrial data into fundamental and inference metadata is motivated by the subspace decomposition technique used in the C2 paradigm. In C2, the data is decomposed into its observable and non-observable space spanned by the dominant "active" DOFs and non-dominant "noisy" DOFs respectively. The observable subspace carries information on the dominant behavior of the process variables in the data which are determined by the underlying physical model, operational conditions, process parameters etc. whereas the non-observable space contains information about most of the noise in the model. The above decomposition is extended further in the DIOD work by further decomposing the observable space into the fundamental and inference metadata. As mentioned in the previous section, the fundamental metadata contain information pertaining to the physical system such as the underlying differential equations whereas the inference metadata contain information on the process parameters, boundary conditions etc. This decomposition may be achieved through ROM methods such as proper orthogonal decomposition, kernel PCA , etc. Given a proprietary dataset represented using the multivariate function $y(\mathbf{x}, \boldsymbol{\alpha})$, the goal of the DIOD paradigm is to decompose this function into the fundamental metadata, represented by the basis functions $\psi_i(\mathbf{x})$, and the inference metadata $\phi_i(\boldsymbol{\alpha})$. The decomposition may be represented as a k-rank approximation depending on the user-defined tolerance ϵ as demonstrated in Eqs. 4.1 and 4.2.

$$y(\mathbf{x}, \boldsymbol{\alpha}) \approx \sum_{i}^{k} \psi_i(\mathbf{x})\phi_i(\boldsymbol{\alpha}) \tag{4.1}$$

$$\left| y(\mathbf{x}, \boldsymbol{\alpha}) - \sum_{i}^{k} \psi_i(\mathbf{x})\phi_i(\boldsymbol{\alpha}) \right| < \epsilon \tag{4.2}$$

Here, the variable $\mathbf{x}$ depends on the target application. For example, it may represent the pixel location on an image or the spatial/temporal modes of evolution in the data (exponential, polynomial etc.). The variable $\boldsymbol{\alpha}$ is the chief source of variation among the samples in a dataset. It may represent varying boundary conditions, process parameters, material properties, etc. that must be extracted for inferential applications. The following sections describe the masking methodology in further detail for the fundamental and inference metadata.

5.2.1 Data Masking: Fundamental Metadata

In this section, the fundamental metadata of the proprietary dataset is masked, while the inference metadata is preserved. Here, the key assumption is that no information about the system identity may be gleaned from the inference metadata, and no further assumptions are made about the target AI/ML application. The proposed methodology may be used to create datasets to benchmark various AI/ML tools since an ideal AI/ML tool is expected to be invariant to the fundamental metadata and act on the inference metadata only.

The first step of the DIOD paradigm is to perform the above decomposition using established ROM techniques, among which a gradient-based active subspace identification technique is described below as an example. Via random perturbations of the various input parameters to a system model, it is possible to compute the gradient of the function $y(\mathbf{x}, \boldsymbol{\alpha})$ and find the active subspace through PCA of the gradient matrix. Geometrically, the principal features here denote the directions of maximum variance, i.e., they describe the directions of input perturbations that have the maximum impact on the system responses. The space orthogonal to the active subspace is expected to have minimal impact on the system and may thus be discarded for the purposes of DIOD. Note that this subspace, termed the non-observable space, plays a key role in the C2 paradigm as courier variables. The principal features, describing the possible modes of evolution of the response, are determined by the underlying differential equations of the system. On the other hand, the prominence of these features, represented by the dot product of the features with a given sample from the dataset, depends on the process parameters used in generating said sample. Therefore, in this mode of separation, the principal features are a representation of the fundamental metadata, while the corresponding dot products, called feature components, are a representation of the inference metadata. These components are useful for inferential purposes such as anomaly detection, optimization, etc. akin to the concept of LOCs and HOCs described in section 2.4.12.1 .

The second step involves the development of a library of concealment operators to mask the extracted fundamental metadata while preserving the inference metadata. Using similar ROM-based decomposition techniques, the fundamental metadata of multiple generic systems, which are often well-understood, may be extracted to develop a deception kernel for the purposes of concealment. The goal of the deception kernel is to replace the fundamental metadata of the proprietary system with that of a generic system; however, it is critical that the kernel is invertible

and only known to the owner of the proprietary system. Here, the obfuscation through the kernel is a "permanent" mask in the sense that any associations with the proprietary system are permanently altered. The decomposition step and the development of the library for various generic systems are a one-time cost to the proprietor owner, and DIOD datasets may be generated by directly implementing the fusion step outlined below.

The third step involves the fusion of the inference metadata of the proprietary system and the fundamental metadata of the generic system. The concealment operator acts on the proprietary dataset $y(\mathbf{x}, \boldsymbol{\alpha})$ by fusing its inference metadata $\phi_i(\boldsymbol{\alpha})$ onto the fundamental metadata of the generic system $\varphi_i(\mathbf{x}')$ to generate the DIOD dataset $y'(\mathbf{x}, \boldsymbol{\alpha})$ using the deception kernel $K(\mathbf{x}', \boldsymbol{x})$ as shown in Eqs. 4.3 and 4.4. Here the ψ_i^* operator represents the transpose operator with the property that the product $\psi_i^* \psi_j$ is identity when $i = j$ and zero otherwise, i.e., the basis functions are orthonormal.

$$K(\mathbf{x}', \mathbf{x}) = \sum_{i=1}^{k} \varphi_i(\mathbf{x}') \psi_i^*(\mathbf{x}) \tag{4.3}$$

$$y'(\mathbf{x}', \boldsymbol{\alpha}) = K(\mathbf{x}', \mathbf{x})\, y(\mathbf{x}, \boldsymbol{\alpha}) \tag{4.4}$$

Since the deception kernel is invertible, any inference made on the DIOD datasets may be traced back to the original dataset. To further illustrate this condition, consider a classifier C that trained on $y(\mathbf{x}, \boldsymbol{\alpha})$ or $y'(\mathbf{x}', \boldsymbol{\alpha})$ attempting to classify new data z or its DIOD version z' respectively. The DIOD paradigm requires that Eq. 4.5 is satisfied.

$$C_{y(\mathbf{x}, \boldsymbol{\alpha})}(z) = C_{y'(\mathbf{x}', \boldsymbol{\alpha})}(z') \tag{4.5}$$

From the above equations, it may be gleaned that any attempts to reverse-engineer the data through ROM-based techniques leads back to the fundamental metadata of the generic system, $\varphi_i(\mathbf{x}')$, and the inference metadata of the proprietary system, $\phi_i(\boldsymbol{\alpha})$, implying that the inferential properties of the data is preserved and the association with the proprietary system is masked, thus fulfilling the goals of the DIOD paradigm. Section demonstrates the ability of the DIOD paradigm to preserve inference metadata for a wide variety of AI/ML applications using a nuclear reactor and a DCPM simulation in Dymola, thus serving as a benchmark dataset. The following section

provides a mathematical framework for the case where the inference metadata may also need to be obfuscated to be more representative of the inference metadata commonly associated with the generic system.

5.2.2 Data Masking: Inference Metadata

This section addresses the concern that direct fusion of the inference metadata may be insufficient in protecting system identity since some industrial systems may have process parameters, boundary conditions etc. that are unique to the system. There is thus a need for further obfuscation of the inference metadata, which is a challenging task given that the data must remain usable for the target AI/ML application. Here, the deception kernel is modified by fine-tuning it to the target AI/ML application to render the inference metadata more representative of the generic system. This is done through the concept of mutual information, discussed in further detail below.

The mutual information between two variables is a statistical measure that captures all linear and non-linear dependencies between them unlike simple linear correlation [171]–[174]. Derived from the principles of entropy, it represents the gain in information of one variable with the knowledge of the other, and vice versa, making it a symmetric measure. For example, suppose a coin is tossed twice and classified as 1 if the resulting faces are the same, and 0 otherwise. The initial uncertainty/entropy in the classifier label is 50% (1 bit) since the two outcomes are equally likely. On revealing the face on only one coin, it is observed the entropy in the label is still 1 bit, implying that no information was gained. Intuitively, it may be ascertained that the mutual information between the classifier label and revealing the result of one coin toss is 0. However, if both coins are revealed, the label is known with complete certainty, i.e., it has no entropy, resulting in an information gain. Therefore, the mutual information between the classifier label and revealing the result on both coins is the corresponding reduction in entropy, i.e., 1 bit. Mathematically, the mutual information $I(a;b)$ between two variables a and b given the entropy function $H(\cdot)$ is given by $I(a;b) = H(a) - H(a|b)$.

In the context of classification and DIOD, the mutual information between the inference metadata and the classifier label represents the highest possible separability of the dataset using only the inference metadata [170]. This has implications in feature selection for AI/ML applications where additional features that do not impact the mutual information of the label may be discarded as redundant. In the DIOD paradigm, this means that not all the information in the

inference metadata is necessary, and subsequently some may be discarded, freeing up additional DOFs to introduce extraneous correlations in the inference metadata that fit the requirements of the generic system. Furthermore, mutual information is unaffected by invertible transformations that preserve the dimensionality of the inference metadata, allowing for further obfuscation. This invariance to invertible transformations and the addition of extraneous variables is described using Eq. 4.6.

$$I\big(L\big(y(\mathbf{x},\boldsymbol{\alpha})\big);[\phi_i(\boldsymbol{\alpha}) \quad \xi]\big) = I\big(L\big(y(\mathbf{x},\boldsymbol{\alpha})\big);\phi_i(\boldsymbol{\alpha})\big) = I\left(L\big(y(\mathbf{x},\boldsymbol{\alpha})\big);f\big(\phi_i(\boldsymbol{\alpha})\big)\right) \tag{4.6}$$

Here, the function $L(\cdot)$ is the classifier label corresponding to the data in $y(\mathbf{x},\boldsymbol{\alpha})$, $f(\cdot)$ is assumed to be an invertible function, and the extraneous variable ξ has no impact on the classifier label. Using this framework, it can be readily seen that simply masking the fundamental metadata as done in section 4.2.1 does not affect the performance of AI/ML applications since the mutual information between the label and the DIOD dataset is the same as that between the label and the original dataset. This result may be obtained by setting $\xi = \psi_i(\mathbf{x})$ or $\xi = \varphi(\mathbf{x}')$ and $f\big(\phi_i(\boldsymbol{\alpha})\big) = K(\mathbf{x}',\mathbf{x})\psi_i(\mathbf{x})\phi_i(\boldsymbol{\alpha}) = y'(\mathbf{x}',\boldsymbol{\alpha})$. Additionally, in the previous implementation where the inference metadata was preserved, it is observed that there exists an invertible transformation between $y'(\mathbf{x}',\boldsymbol{\alpha})$ and $y(\mathbf{x},\boldsymbol{\alpha})$ through the deception kernel, leading to the same conclusion that any inference made using either dataset must be identical.

While $\boldsymbol{\alpha}$ may be thought of as process parameters, boundary conditions etc., in the context of AI/ML algorithms, it may also be thought of as the latent variables of the dataset $y(\mathbf{x},\boldsymbol{\alpha})$ that determine the separability of the classification problem. As long as there is an invertible transformation from the dataset $y(\mathbf{x},\boldsymbol{\alpha})$ and its latent variables $\boldsymbol{\alpha}$, no mutual information is lost in the reduction. The latent space may be identified through autoencoders, parameter extraction tools like physics-informed neural networks, reverse-engineering tools such as SINDy and PDE-FIND etc. Further transformations may be performed on these variables to generate another set of latent variables, which may be fused with the fundamental metadata of the generic system to create the DIOD dataset. Multiple levels of masking may be achieved using this technique depending on the classification task, as explored below.

The first level of masking, as done in the previous section, only permits transformations of the inference metadata that preserve or increase their dimensionality. This is necessary to allow for a transformation to the proprietary dataset $y(\mathbf{x}, \boldsymbol{\alpha})$ from the DIOD dataset $y'(\mathbf{x}', \boldsymbol{\alpha})$ via the deception kernel, which in turn preserves the mutual information as shown in Eq. 4.7. This may be suitable for cases where the target AI/ML application is unknown, and all the inference metadata must be preserved. However, this level of masking is limited in its applicability and requires careful consideration of the generic system since the latter requires at least as many dimensions as that of the inference metadata.

$$I\big(y(\mathbf{x}, \boldsymbol{\alpha}); \phi_i(\boldsymbol{\alpha})\big) = I\left(y'(\mathbf{x}', \boldsymbol{\alpha}); f\big(\phi_i(\boldsymbol{\alpha})\big)\right) \tag{4.7}$$

The second level of masking may be permitted when the inference metadata itself is reducible to its latent space described by $\boldsymbol{\alpha}$. Here, the above constraint may be relaxed to preserving or increasing the dimensionality of the latent space instead since there exists a transformation to $y(\mathbf{x}, \boldsymbol{\alpha})$ from $y'(\mathbf{x}', \boldsymbol{\alpha})$. The transformation in this case is a composition of two transformations, namely, one from the dataset to the inference metadata, and another from the inference metadata to the latent space. Nevertheless, the mutual information between the proprietary dataset and its latent space, and that between the DIOD dataset and its latent space are preserved, thus ensuring similar inference on both datasets as shown in Eq. 4.8. This is the limit of masking achievable for unsupervised learning applications.

$$I(y(\mathbf{x}, \boldsymbol{\alpha}); \boldsymbol{\alpha}) = I\left(y'\big((\mathbf{x}', \boldsymbol{\alpha})\big); f(\boldsymbol{\alpha})\right) \tag{4.8}$$

The third level of masking is permitted when the target AI/ML application is a supervised problem, e.g., classification with labels $L(\cdot)$ known a priori. Here, the requirement of a transformation to the proprietary dataset $y(\mathbf{x}, \boldsymbol{\alpha})$ from the DIOD dataset $y'(\mathbf{x}', \boldsymbol{\alpha})$ may be relaxed since not all latent variables may be relevant for the classification problem. While preserving the subset of relevant variables $\boldsymbol{\beta}$ and discarding the rest does not permit a transformation to $y(\mathbf{x}, \boldsymbol{\alpha})$ from $y'(\mathbf{x}', \boldsymbol{\alpha})$, it does not affect the ability of the classifier since the discarded variables are extraneous and do not affect the mutual information between the label and the relevant latent

variables as shown in Eq. 4.9. In this case, only the dimensionality of the relevant variables needs to be preserved as the class labels of a proprietary dataset and its corresponding DIOD version are the same. The above discussion is summarized in Table 3.

$$I\big(L\big(y(\mathbf{x},\boldsymbol{\alpha})\big);\boldsymbol{\beta}\big) = I\left(L\big(y'(\mathbf{x}',\boldsymbol{\alpha})\big);f(\boldsymbol{\beta})\right) \tag{4.9}$$

Table 3: Levels of masking of inference metadata

Masking Level	Constraint	Explanation
1	$I\big(y(\mathbf{x},\boldsymbol{\alpha});\phi_i(\boldsymbol{\alpha})\big) = I\left(y'(\mathbf{x}',\boldsymbol{\alpha});f\big(\phi_i(\boldsymbol{\alpha})\big)\right)$	Transformations on $\phi_i(\alpha)$ must preserve their dimensionality where f is invertible. Requires transformation from $y' \rightarrow y$
2	$I(y(\mathbf{x},\boldsymbol{\alpha});\boldsymbol{\alpha}) = I\left(y'\big((\mathbf{x}',\boldsymbol{\alpha})\big);f(\boldsymbol{\alpha})\right)$	Transformations on α must preserve its dimensionality where f is invertible. Requires transformation from $y' \rightarrow y$
3	$I\big(L\big(y(\mathbf{x},\boldsymbol{\alpha})\big);\boldsymbol{\beta}\big) = I\left(L\big(y'(\mathbf{x}',\boldsymbol{\alpha})\big);f(\boldsymbol{\beta})\right)$	Transformations on α must preserve the dimensionality of the relevant subset β where f is invertible

Note that in all the above cases, extraneous variables may be added to the inference metadata, the latent variables, or the relevant subset of latent variables without losing any information. This implies that the dimensionality of the generic system needs to be at least equal to that of the inference metadata, the latent variables, or the relevant subset depending on the desired level of masking. Of the three, the least strict criterion is Eq. 5.9 and it allows for the deepest level of masking fine-tuned to the target AI/ML application. This is illustrated through the example below.

Consider a proprietary system simulated over different conditions by varying its m process parameters to generate the proprietary dataset and a generic system that requires n constraints to ensure that the inference metadata is representative. Here, level 1 masking attempts to preserve the dimensionality m of the process parameters/inference metadata of the proprietary system and requires the generic system to have at least $m + n$ dimensions available. Next, suppose the m

process parameters are spanned by a latent space of dimension $r < m$. This frees up $m - r$ DOFs to implement some constraints of the generic system, implying that $r + n$ dimensions are sufficient for the generic system, thus achieving level 2 masking. Lastly, if the target AI/ML task is to build a classifier that only requires $s < r$-dimensional subset to characterize the class label, $m - s$ constraints of the generic system may be implemented. Here, level 3 masking may be achieved if the generic system has at least $s + n$ dimensions. The best-case scenario is achieved if the dimension of the relevant latent variables is $s = m - n$, fulfilling all the constraints of the generic system.

One class of reverse-engineering tools used in the present work is a set of data-driven techniques to extract underlying ODEs and PDEs [13], [51], [52], [158] of nonlinear dynamical systems in the applied mathematics community. In these methods, the challenge of finding the underlying system of ODEs/PDEs is transformed into a regression problem, where the data is fit to a system of differential equations consisting of polynomials, sines, cosines, partial derivatives, higher-order derivatives etc. In the SINDy and PDE-find algorithms proposed in [51], [52] for example, sparse regression techniques combined with Pareto analysis is used to avoid overfitting the data and obtain parsimonious models that offer the best balance between accuracy and complexity. Additionally, in PDE-FIND, the data points considered to build the model are randomly subsampled from the whole dataset with good approximations resulting with just 2-3% subsampling. The methodology is described using Eqs. 4.10 and 4.11 below.

$$\frac{\partial y(x,\alpha)}{\partial t} = f\left(\alpha, y, y^2, x, x^2, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \dots\right) \tag{4.10}$$

$$\mathbf{Y}_t = \boldsymbol{\Psi}(\mathbf{Y}, \mathbf{Q})\, \boldsymbol{\Phi}(\boldsymbol{\alpha}) \tag{4.11}$$

In Eq.4.11, which is the matrix analog of Eq. 4.1, $\mathbf{Y}$ denotes the data matrix subsampled at various points in space-time, $\mathbf{Y}_t$ denotes the time-derivative of the data at these points, $\mathbf{Q}$ denotes any additional forcing functions, $\boldsymbol{\Psi}$ denotes the library of fitting functions considered, i.e., the fundamental metadata, and $\boldsymbol{\Phi}$ represents the coefficients, i.e. the inference metadata, to be determined through sparse regression. While the derivatives may be approximated using simple finite difference methods, it may be necessary to use smoothing or polynomial interpolation for noisy data. Although the model may be rendered resilient to noise using interpolation,

preprocessing the data to decrease noise using denoising techniques such as singular value decomposition [175], singular spectrum analysis [176], [177] etc. may be considered.

Section 4.4 demonstrates the ability of DIOD to leverage reverse-engineering tools to prevent reverse-engineering of proprietary datasets and mask both the fundamental and inference metadata. A nuclear reactor dataset is successfully transformed into that from a nonlinear spring-mass system with various levels of masking achieved depending on the target AI/ML application.

5.3 Data Masking for AI/ML Benchmark Datasets

In this section, the DIOD methodology is implemented using the simulation of a nuclear reactor and a DCPM representing a proprietary and generic system respectively for data masking. The inlet temperature of the reactor and the current output are the measured quantities, and the above simulations are carried out in Dymola (version 2020x) as shown in Fig 1 below [110], [111]. The Westinghouse 4-Loop PWR example from the TRANSFORM package [113] and the current-controlled DCPM from the default Modelica package are simulated under various operating conditions. The two processes are simulated until steady state for 100 seconds. In this implementation, the inlet temperature represents sensitive information that must be masked since it can be linked to a nuclear reactor based on its temporal evolution by a knowledgeable adversary. However, the data must be masked in a manner that preserves the AI/ML-relevant inference metadata as obtained from the decomposition in Eq. 4.1. To this end, the current-controlled DCPM serves as an ideal canvas to fuse the inference metadata since it is representative of a non-critical, generic, and well-understood system. The goal of the DIOD methodology is to fuse the inference data from the PWR with the fundamental metadata of the DCPM to create the DIOD version of the data. This is accomplished using Eqs. 4.3 and 4.4 via kernel deception using the concealment operators developed for the DCPM system as shown in Figure 17.

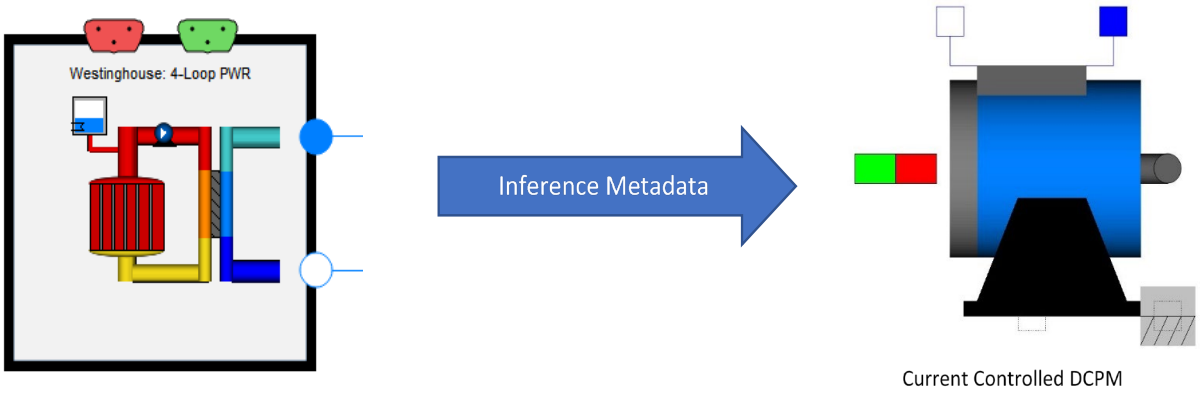

Figure 30: Goal of DIOD paradigm

In the following sections, the DIOD methodology is implemented and validated via statistical and machine-learning techniques. They are organized as follows. In section 4.3.1, the inference metadata from the PWR under two different simulation conditions are concealed within the DCPM responses. The validation process is performed using simple statistical tools to ensure that in-class and between-class separability are maintained. In section 4.3.2, we assumed that the target application does not have any pre-existing knowledge of the dataset and implemented an unsupervised k-means algorithm to simulate the inference of AI/ML tools. The goal is to ensure that the same separability exists in the inference metadata of the PWR and the DIOD data. In section 4.3.3, a more complex analysis, singular value decomposition, is performed to extract information about both the fundamental and inference metadata of the system. The validation process ensures that the DIOD data and the PWR data share the same inference metadata but different fundamental metadata to conceal proprietary information and verifies this using the mutual information of the response and the inference metadata. Lastly, in section 4.3.4, two responses from each system are considered and the correlation between these responses is computed to produce correlation curves that characterize the fundamental metadata of the system. Here, the DIOD methodology is validated by ensuring that the DIOD correlation curve exhibits significantly different behavior than that of the PWR.

5.3.1 Domain Knowledge

In this section, the DIOD methodology is demonstrated by fusing the inference metadata of the PWR simulations with the fundamental metadata of the current-controlled DCPM as seen in Figure 18. The inlet temperature of the PWR is simulated under two different conditions, namely a) normal operation and b) partial pump failure as shown on the left subplot of Fig. 3. For each condition, five datasets were generated by varying the mass flow rates and the final pump revolutions per minute (RPM), respectively, although only one set from each condition is plotted for visual acuity. The goal of the manuscript is to anonymize the above data by fusing the inference metadata of the two operating conditions with the current-controlled DCPM simulation as seen on the right subplot of Figure 18. Figure 19 depicts the DIOD version of the inlet temperature data from Figure 18 after applying the methodology as described by Eqs. 4.3 and 4.4.

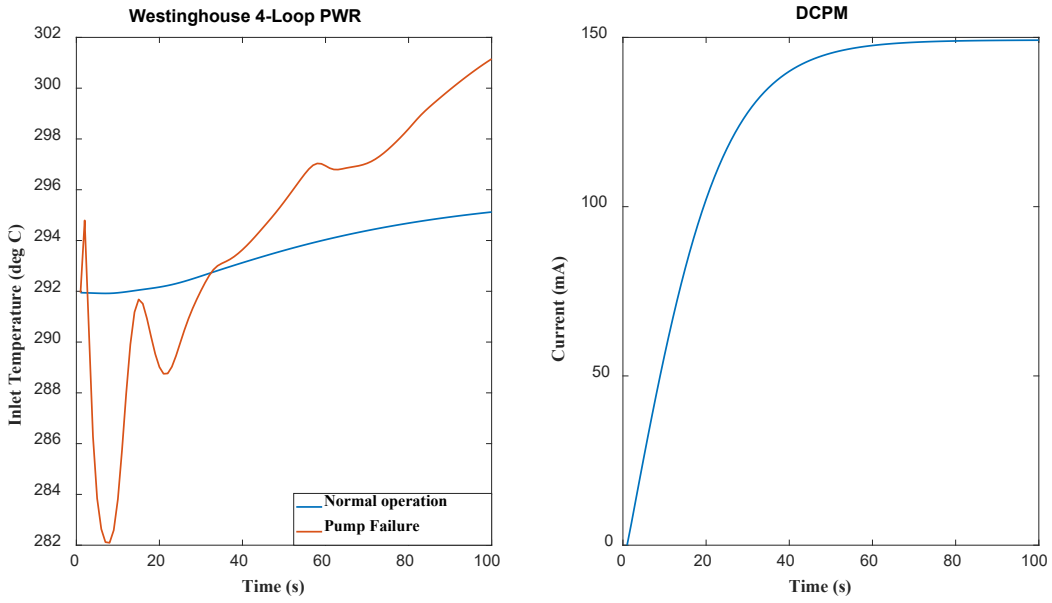

Figure 31: Representative data from proprietary (PWR) and generic (DCPM) system

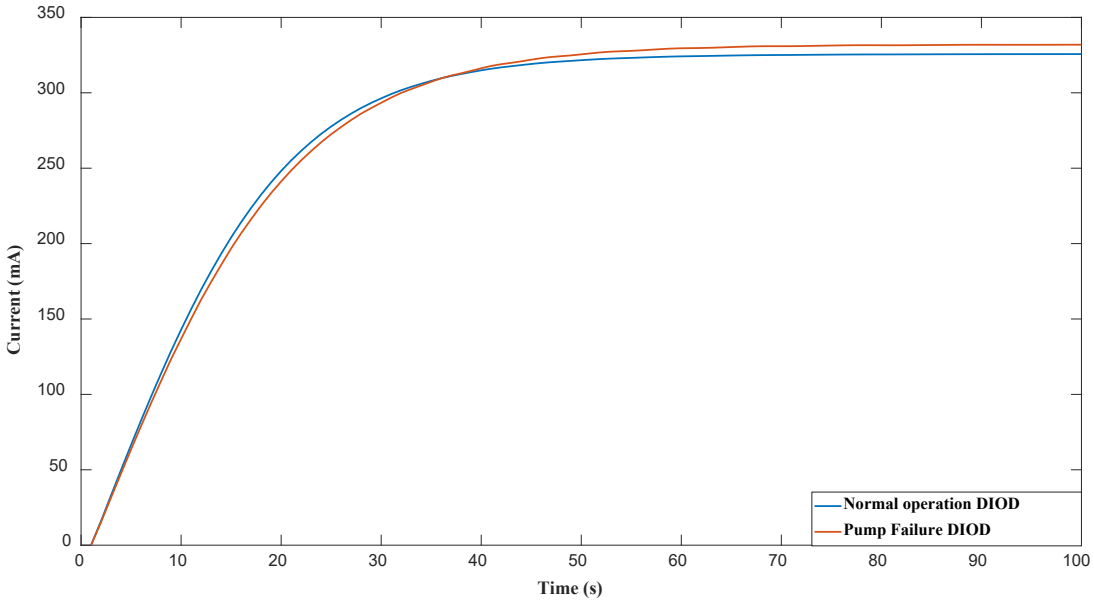

Figure 32: DIOD version of PWR data

The DIOD version of the data is similar in appearance to the operation characteristic of the current-controlled DCPM, and the visual separability is maintained within the two operating conditions—partial pump failure and normal operation. Using domain knowledge that the DCPM current follows an exponential profile, statistical analysis reveals the separation of the two classes in the DIOD data via a significantly different time constant. Additionally, the in-class separability in each class is in the final saturation value of the DIOD curves as tabulated in Table 4. The fraction of pump power at steady state after partial pump failure is denoted by μ.

Table 4: Classification based on domain knowledge

Operation Mode	Normal Operation Mass Flow Rate (kg/s)					Partial Pump Failure (μ)				
	450	460	470	480	490	0.2	0.25	0.3	0.35	0.4
DIOD time constant (s)	16	16	16	16	16	17	17	17	17	17
DIOD sat. current (mA)	327.8	327.2	326.7	326.2	325.7	331.9	331.4	330.8	329.6	329.2

5.3.2 Unsupervised AI/ML

The next task is to use an AI/ML tool to classify the 10 datasets based on their operating conditions (i.e., normal operation and partial pump failure without domain knowledge). In this section, we assumed that there is no a priori knowledge of the operating conditions/classes and utilized an unsupervised learning algorithm to cluster the datasets. The PWR inlet temperature is simulated under normal operating conditions and partial pump failures by varying the mass flow rate and fraction of pump power at steady state, respectively, as shown in section 4.3.1. The resultant time-series are separated into two clusters using a k-means clustering algorithm without explicitly labeling the membership class of each dataset or the number of members in each class (unsupervised). Table 5 shows that the two clusters correspond to the normal operation and partial pump failure conditions. This can be attributed to the vastly different structure of the two curves as seen on the left subplot of Figure 18 above.

Then, the DIOD methodology is applied onto the inlet temperature data using the fundamental metadata from the DCPM. The above process is repeated and the k-means clustering algorithm is applied on the DIOD datasets. The class separability is still maintained in the DIOD datasets and corresponds to the normal operation and partial pump failure conditions. Table 5 tabulates the results and demonstrates that the class separability is maintained even in an unsupervised learning environment where we assumed that the tool has no domain knowledge of the system.

Table 5: Classification using k-means clustering

Operation Mode	Normal Operation Mass Flow Rate (kg/s)					Partial Pump Failure (μ)				
	450	460	470	480	490	0.2	0.25	0.3	0.35	0.4
Cluster index – PWR data	1	1	1	1	1	2	2	2	2	2
Cluster index – DIOD data	1	1	1	1	1	2	2	2	2	2

5.3.3 Singular Value Decomposition

In previous sections, the extraction of inference metadata and their separability was demonstrated using statistical and AI/ML tools. In this section, we use singular value decomposition (SVD) to extract information about both the fundamental and inference metadata

from the PWR, DCPM, and DIOD data. SVD is a widely used analysis tool and is defined via an orthogonal transformation of the data onto a new set of coordinate axes ordered from greatest to least variance. The given dataset $\mathbf{z}$ is decomposed as $\mathbf{z} \approx \beta_1 \mathbf{u}_1 + \beta_2 \mathbf{u}_2 + \cdots + \beta_n \mathbf{u}_n$, where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ describe the n dominant features of the dataset and $\beta_1, \beta_2, \ldots, \beta_n$ describe their respective coefficients. The extracted $\mathbf{u}_i$ vectors describe the behavior of the physical system itself and provide some insight into the fundamental metadata of the system. Their coefficients β_i describe the operation of the system and provide information about the inference metadata of the system. The DIOD methodology is validated by preserving existing correlations among the coefficients β_i of the PWR and masking the fundamental metadata of the PWR.

Consider 10 time-series generated by varying the mass flow rate of the PWR under normal operation. As in the previous sections, Eqs. 4.1-4.4 are used to fuse the inference data of the PWR with the DCPM using the library of concealment operators developed earlier, thus generating the DIOD version of the PWR data. Consequently, the PWR data and the DIOD data are decomposed, and the first three coefficients β_1, β_2, and β_3 representing the dominant coefficients are analyzed. Figure 20 presents the correlation between the coefficients β_i obtained from the SVD of the PWR and DIOD data and shows that the correlations are preserved across the PWR and the DIOD datasets.

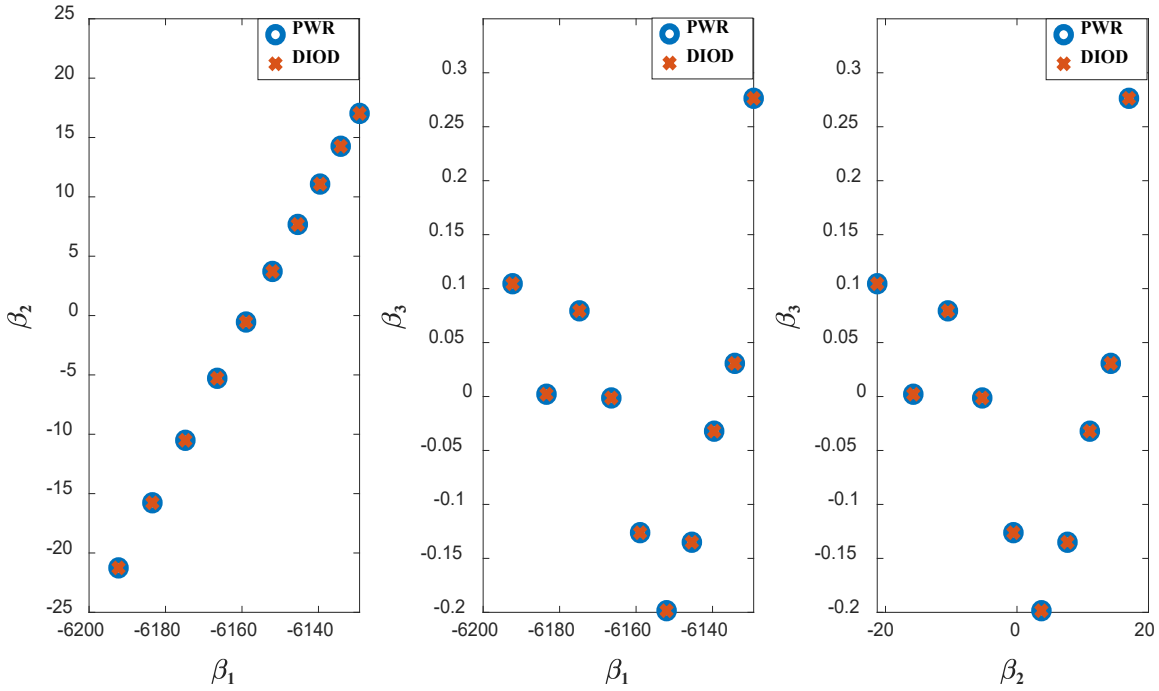

Figure 33: Correlation among SVD coefficients

Next, we consider the first three dominant **u** vectors obtained from the decomposition of the PWR and its DIOD version as shown in Figure 21. A successful implementation of the DIOD methodology completely masks the fundamental metadata of the proprietary system (PWR). The **u** vectors of the PWR containing information about the fundamental metadata do not resemble those extracted from the DIOD version, thus protecting the fundamental metadata of the PWR. Additionally, the extracted vectors of the DIOD data resemble those of the DCPM, implying that the underlying physical processes of the two are similar. Therefore, any analysis of the **u** vectors of the DIOD dataset provides insight into the generic DCPM system and not the proprietary PWR.

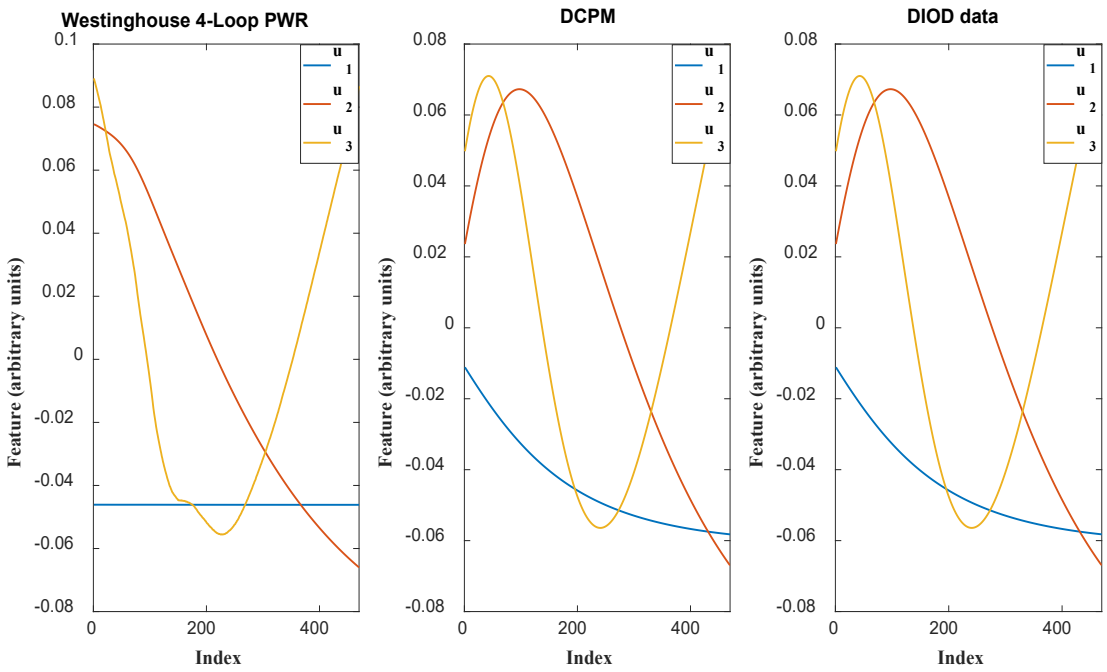

Figure 34: **u** vectors from SVD

5.3.4 Response Correlation

In the previous section, the DIOD methodology was validated using SVD by preserving the correlations among the SVD coefficients of the responses representing inference metadata and masking the fundamental metadata of the PWR. In this section, the masking of fundamental metadata of the PWR is further demonstrated using the correlations among the responses themselves. Every physical system is expected to have its own set of correlations among its responses based on the underlying physical model. For example, the data from an experiment on a resistor may exhibit a linear relationship between the current and the voltage. A successful DIOD implementation masks these relationships so that any attempts at inference do not lead back to the original proprietary system. In this experiment, two responses from each system are considered—namely, the inlet and outlet temperature from the PWR and the current and rotation speed of the DCPM. The correlation between the inlet and outlet temperature is computed to produce correlation curves that characterize the fundamental metadata of the PWR.

Consider a DIOD implementation involving the above responses from the Westinghouse 4-Loop PWR and current-controlled DCPM system described at the beginning of 4.3. The PWR and the DCPM are simulated until steady state is achieved. Using Eqs. 4.1-4.4, the inference

metadata of the outlet temperature is fused with the speed data, while that of the inlet temperature is fused with the current data.

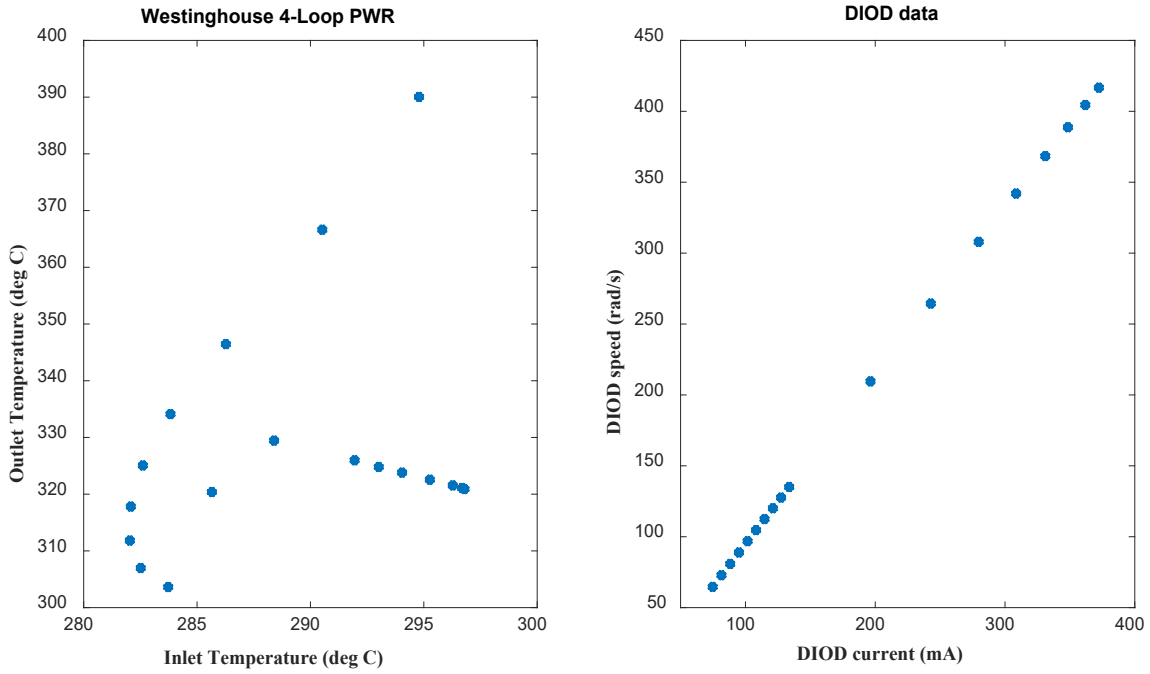

Figure 35: Response Correlation

The correlation curves of the PWR and DIOD datasets are shown for a representative case in Fig. 7 using the responses obtained from each system. The DIOD methodology is validated by observing that the correlation curve of the Westinghouse PWR is masked in the DIOD data, thus protecting it from discovery. However, the idea may be further extended by using invertible mathematical transformations to better fit the features of the generic system depending on the target application. For example, the DIOD inference metadata could be scaled/transformed to exhibit a relationship similar to a DCPM correlation curve. The DIOD data can thus be reasonably expected to have come from the DCPM system, achieving another level of masking if desired. This is desirable for reverse-engineering applications where the invariance of the mutual information may be exploited for further masking, as demonstrated in the following section.

5.4 Data Masking against Reverse-Engineering

In this section, we consider a nuclear reactor, representing the proprietary system, and a simple spring-mass system, representing the generic system, on which the DIOD methodology is to be implemented. The reactor physics is described by the single-group point-kinetics model in Eq. 4.12, while the spring-mass system is described for a non-linear hardening/softening spring as described in Eq. 4.13.

$$\begin{aligned} \dot{P} &= \frac{(\rho - \beta)}{\Lambda} P + \lambda C \\ \dot{C} &= \frac{\beta}{\Lambda} P - \lambda C \end{aligned} \tag{4.12}$$

In the above expression, P is the power of the reactor with the $\dot{}$ operator denoting its first derivative with respect to time, ρ is the initial reactivity inserted into the system, Λ is the prompt generation time of the neutrons, λ is the decay constant corresponding to one group, and β is the delayed neutron fraction. These parameters vary depending on the experimental conditions of the reactor such as its geometry, fuel enrichment, fuel composition etc. For the purposes of this manuscript, ρ is held to be a fixed quantity, i.e., we would like to see the effect of inserting a small amount of reactivity over time into various reactor compositions.

$$m\ddot{x} + l\dot{x} + kx + nx^3 = 0 \tag{4.13}$$

In Eq. 4.13, x is the position of the spring with the $\dot{}$ and $\ddot{}$ operators denoting its first and second derivative with respect to time respectively, m is the mass of the object attached to the spring, l is the damping coefficient of the dashpot, k and n are parameters describing the stiffness of a nonlinear spring. These parameters may vary depending on the material used for the spring and the dashpot. The following subsections illustrate the constraints imposed on the spring-mass system for the three different masking levels.

For the purposes of this manuscript and to demonstrate the separability criterion, it is assumed that the labels are known and binary, i.e., the problem is set up as a balanced supervised binary classification problem. Additionally, the binary labels only depend on the latent variables β and Λ. The classification criteria are shown in Algorithm **1** where $\mathrm{U}(a, b)$ denotes a uniform

distribution on the interval $[a, b] \in \mathbf{R}$. Algorithm **1** divides the dataset such that 40% of the data belongs to the class corresponding to label 0, 40% of the data belongs to the class corresponding to label 1, and 20% of the data is randomly split between the two labels. Therefore, the maximum achievable accuracy for the optimal classifier is 90%. A total of 10,000 samples were generated for the reactor data and labeled, of which 2,000 samples were selected as test data for the neural network.

Algorithm 1:	Data classification
Input:	Delayed neutron fraction, $\beta \sim \mathrm{U}(0.006, 0.007)$ Prompt generation time, $\Lambda \sim \mathrm{U}(1 * 10^{-5}, 5 * 10^{-5})$ Number of samples in dataset, n
Output:	Label L corresponding to reactor data y using β and Λ
Initialize:	Counter variable $p = 1$
WHILE	$p \leq n$
IF	$4\beta + 100\Lambda \leq 2.85777 * 10^{-2}$
	$L(y) \leftarrow 1$
ELSE IF	$4\beta + 100\Lambda \geq 2.94223 * 10^{-2}$
	$L(y) \leftarrow 0$
ELSE	
	$L(y) \leftarrow$ randint(0, 1)
	$p \leftarrow p + 1$
END	

It is important to note that the algorithm above is only for simulation purposes; in reality, the classification algorithm is unknown while the labels are known and must be devised by the AI/ML researchers using supervised learning algorithms such as neural networks [178], support vector machines [179] etc. on the DIOD data. In the case of unsupervised learning, the highest achievable masking is level 2, and no further reduction is possible since the labels are also unknown and can vary depending on the criteria used. The following subsections implement the different levels of masking on the reactor and the spring-mass system.

5.4.1 Level 1 Masking – Inference Metadata

In this experiment, the set of ODEs describing the reactor kinetics (Eq. 4.12) is extracted using SINDy from the dataset generated from various simulations. The inference metadata of the

reactor are the coefficients of the various terms in the equation. Since the extracted coefficients are quite close to the true value, the DIOD dataset is generated using the true value of the inference metadata. In reality, there may be a minor loss in information due to numerical error or an inability to form a parsimonious model using SINDy.

In this case, the inference metadata consists of four coefficients - $\frac{(\rho-\beta)}{\Lambda}$, $\frac{\beta}{\Lambda}$, λ and $-\lambda$, where ρ is fixed. In level 1 masking, no further reduction based on the latent variables or the classification label is performed, and therefore, the generic spring-mass system requires four parameters. In the nuclear data, the coefficients $\frac{\beta}{\Lambda}$ and λ are non-negative. However, in the spring-mass system, the mass m, the spring constant k, and the damping coefficient l are non-negative, while the stiffness parameter n allows for small negative values to ensure a bounded spring response. Therefore, invertible transformations are employed prior to the generation of the DIOD dataset as shown in the flowchart in Figure 23.

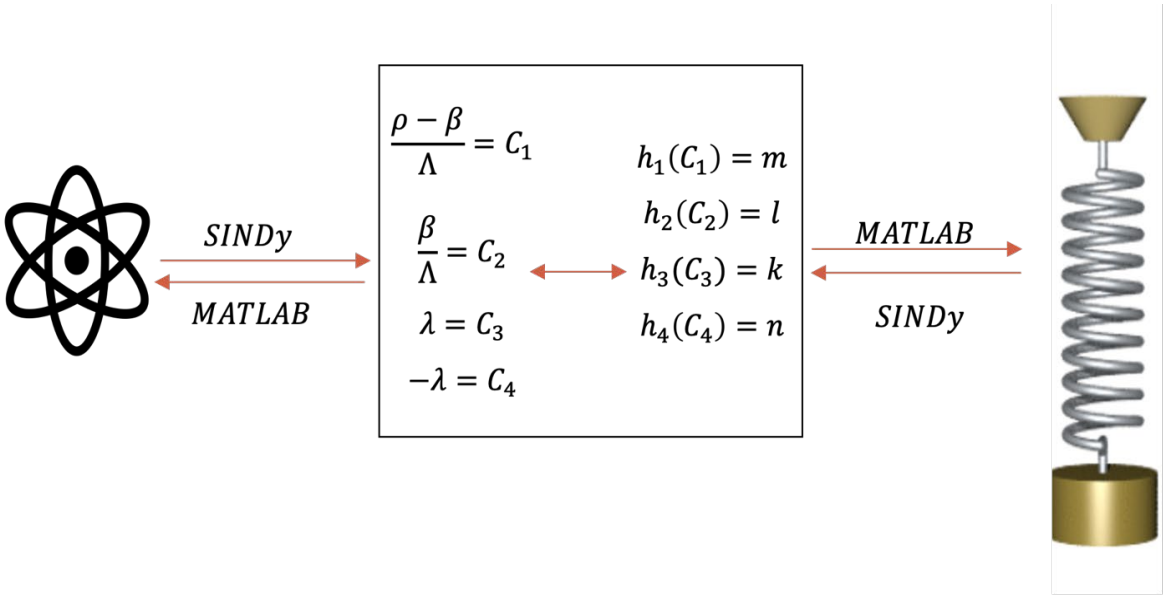

Figure 36: Level 1 Masking

The four arbitrarily chosen transformations used were $h_1(C_1) = -C_1/10$, $h_2(C_2) = e^{C_2/1000}$, $h_3(C_3) = -\ln C_3$, and $h_4(C_4) = e^{C_4} + 7$. Since we have an invertible chain, $y \leftrightarrow \left\{\frac{\rho-\beta}{\Lambda}, \frac{\beta}{\Lambda}, \lambda, -\lambda\right\} \leftrightarrow \{m, l, k, n\} \leftrightarrow y'$, the mutual information between the nuclear data and each of its inference metadata (coefficients) is preserved in the DIOD dataset. A neural network is trained

on the inference metadata of the proprietary system, i.e., the four coefficients $\left\{\frac{\rho-\beta}{\Lambda}, \frac{\beta}{\Lambda}, \lambda, -\lambda\right\}$, and later, on the inference metadata extracted from the DIOD version, i.e., the four coefficients $\{m, l, k, n\}$. It is observed that similar classification results are obtained in both cases as seen in Table 6.

Table 6: Level 1 Classification Results

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Reactor Data	92.7	91.1	91.9
DIOD Data	87.1	92.8	89.8

The sensitivity refers to the percentage of samples in the DIOD dataset having the label 1 that were correctly classified as 1, while the specificity refers to the percentage of samples having the label 0 that were correctly classified as 0. The accuracy is the percentage of samples that were classified to their correct class. By design, the accuracy is expected to be around 90% within error, as observed, while the sensitivity and specificity may vary depending on how the neural network determines its decision boundary. Minor deviations may arise due to the neural network architecture, randomness within the data, training algorithm used, number of samples etc.

Additionally, among the 80% of the reactor data that is clearly separable as defined in Algorithm **1**, their DIOD versions have the same label as desired, i.e., the transformation is isomorphic with respect to the class label for the separable data. In other words, if a test sample in the separable data is classified with the label 1(or 0), its DIOD version is also classified with the label 1(or 0). Among the remaining 20% that are randomly classified into one of the two classes, since they are inseparable by design, their DIOD versions are not isomorphic to the class label. This satisfies the goal of the DIOD methodology in maintaining class separability as illustrated in Figure 24.

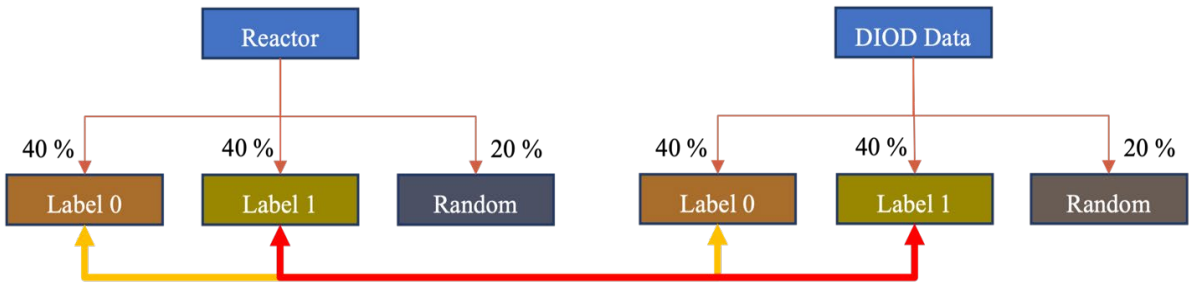

Figure 37: Isomorphism of DIOD transformation to class label

5.4.2 Level 2 Masking – Latent Space

In the above example, one might observe that there are only three variables $\beta, \Lambda,$ and λ and that the inference metadata is a projection of these variables onto a 4-dimensional space. As mentioned in the previous section, the mutual information is invariant to projections onto higher dimensions. Therefore, the generic system need not require four parameters, but three instead to preserve the mutual information. While it is immediately obvious that the third and fourth coefficients are linear transformations of one another in the above implementation, this may not generally be the case, and the problem of finding the latent variables is often solved through more sophisticated methods such as PCA [180], kernel PCA [55], [181], autoencoders [182], [183] etc. The flexibility of the DIOD methodology allows the owner of the proprietary system to control the loss of information (if any) from the reduction of order.

Nevertheless, it is observed that the spring-mass system now has one fewer constraint imposed by the DIOD methodology. For example, the two spring parameters k and n may not be independent in real-life applications since the two are material-dependent properties and relate to the stiffness of the spring. The availability of the extra constraint allows for a functional relationship between the two that may now be respected, thus granting another level of masking. The process is described in Figure 25.

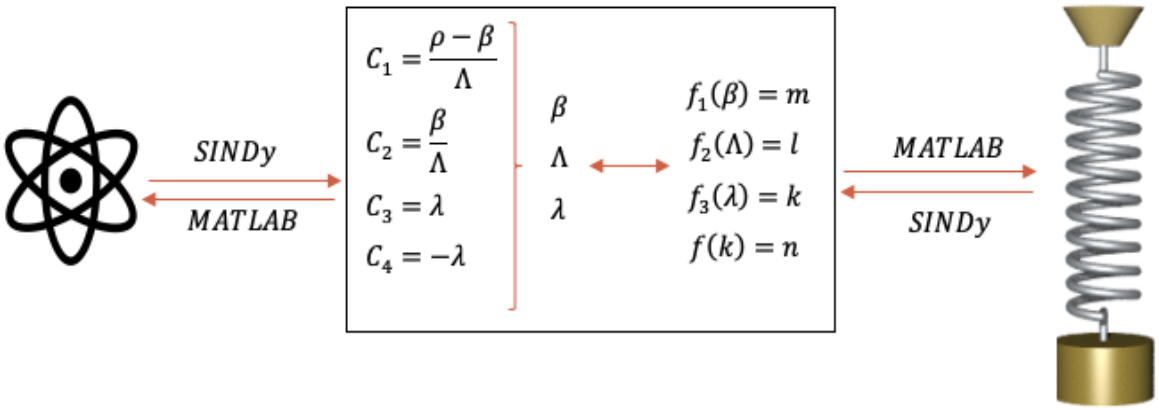

Figure 38: Level 2 Masking

The four arbitrarily chosen transformations used were $f_1(\beta) = e^{-100\beta}$, $f_2(\Lambda) = -\ln \Lambda$, $f_3(\lambda) = 10^5\lambda^3$, and $f(k) = \sqrt{k}$. Similar to the previous section, since we have an invertible chain, $y \leftrightarrow (\beta, \Lambda, \lambda) \leftrightarrow (m, l, k, n) \leftrightarrow y'$, the mutual information between the nuclear data and each of

its latent variables is preserved in the DIOD dataset. A neural network is trained on the latent variables of the proprietary system, i.e., $\{\beta, \Lambda, \lambda\}$, and subsequently on the inference metadata extracted from the DIOD version, i.e., the four coefficients $\{m, l, k, n\}$. It is observed that similar classification results are obtained in both cases as seen in Table 7. Additionally, the isomorphism of the separable data is also observed as in the previous section and illustrated in Figure 24.

Table 7: Level 2 Classification Results

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Reactor Data	90.3	91.9	91.1
DIOD Data	93.5	86.3	89.7

5.4.3 Level 3 Masking – Latent Variables relevant for classification

As mentioned at the beginning of this section, in supervised learning, the dataset is labeled according to some algorithm unknown to the proprietary owner. For the purposes of simulation, the true algorithm is given as a simple partially separable problem in Algorithm **1** where a maximum of 80% true positive and true negative rates may be achieved. Additionally, the true algorithm only uses the latent variables β and Λ and ignores λ, thus providing another level of reduction, but this is also initially unknown to the proprietary owner. What could be done to extract this information?

We refer to 4.9 where the mutual information between the label and the extracted latent variables (via SINDy or other algorithms) is computed. Since the mutual information accounts for all linear and nonlinear dependencies between the variables [170], it is expected that the mutual information between the label and the redundant latent variables is close to zero. In recent decades, techniques such as mutual information neural estimation [171] (MINE) and k-nearest neighbors [172] (kNN) algorithms have been used to estimate the mutual information between two variables, and since the proprietary owner is aware of both the label and the latent space, it is possible to discard the variables irrelevant for classification and achieve the desired level of masking as seen in Figure 26. Here, we reiterate the flexibility of the DIOD methodology in designing constraints based on the user-defined tolerance for error and loss of information.

To verify that the label L and λ have no relationship, we employ Algorithm **1** and compute the mutual information between the two variables using the MINE algorithm, which results in

$I(L(y),[\beta,\Lambda]) = I(L(y),[\beta,\Lambda,\lambda])$, thus verifying the redundancy of the variable λ. Additionally, we also verify that $I(L(y),[\beta,\Lambda]) = 0.78 \approx 0.8$ bits which is the expected mutual information for the problem designed in Algorithm **1**. Intuitively, the algorithm is designed such that 80% of the data can be classified into one of the two labels with complete certainty, while 20% of the data is classified randomly. Since the classification is balanced, the entropy of the label without any knowledge of β and Λ is 1 bit. Therefore, the mutual information, which represents the reduction in the uncertainty of the label after knowledge of β and Λ, is $1 - (0.8 * 0 + 0.2 * 1) = 0.8$ bits. A more in-depth discussion of the relationship between classifiers and mutual information may be found in Refs. [170], [174].

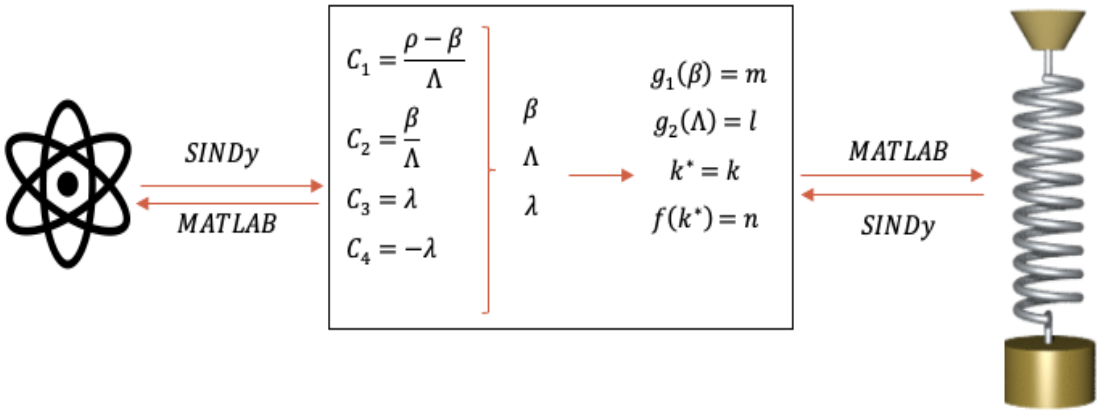

Figure 39: Level 3 Masking

With the additional degree of freedom available, one may impose another constraint required by the generic spring-mass system. For example, the owner of the proprietary data may want to set the value of k and n to a particular material of spring to mask the DIOD data further. In this case, we still have an invertible chain, $L(y) \leftrightarrow (\beta,\Lambda) \leftrightarrow (m,l) \leftrightarrow L(y')$ and thus the mutual information between the label of the nuclear data and the necessary latent variables is preserved in the DIOD dataset.

The four arbitrarily chosen transformations used were $g_1(\beta) = e^{-200\beta}$, $g_2(\Lambda) = 10^5\Lambda$, $k^* = 15$, and $f(k^*) = \sqrt{k} = \sqrt{15}$. A neural network is trained on the required latent variables of the proprietary system, i.e., $\{\beta,\Lambda\}$, and subsequently, on the inference metadata extracted from the DIOD version, i.e., the four coefficients $\{m,l,k,n\}$. It is observed that similar classification results

are obtained in both cases as seen in Table 8. Once again, the DIOD transformation is isomorphic to the label for the separable data as observed in the previous sections and in Figure 24.

Table 8: Level 3 Classification Results

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Reactor Data	91.0	90.4	90.7
DIOD Data	92.1	89.7	90.9

APPENDIX

The appendix provides detailed explanation of the two sub-systems models A for reactor core and B for a steam generator as employed in this work

Reactor Core (Model A):

β: Delayed neutron fraction $= 0.0065$

τ_p: Prompt generation time $= 2.4 * 10^{-4};\ s^{-1}$

λ: Single precursor group decay constant $= 0.08;\ s^{-1}$

α_f: Fuel temperature coefficient $= -2 * 10^{-5};\ ℃^{-1}$

α_m: Moderator temperature coefficient $= -5.3 * 10^{-4};\ ℃^{-1}$

ρ_f: Density of fuel $= 10;\ g\ cm^{-3}$

c_f: Specific heat capacity of fuel $= 0.34;\ J\ g^{-1}\ ℃^{-1}$

A_r: Fuel to cell area ratio $= 0.306796$

V_{cell}: Volume of cell $= 5.74 * 10^7;\ cm^3$

τ_{hx}: Heat exchanger time constant $= 3.54167;\ s^{-1}$

L: Length of fuel rod $= 381;\ cm$

N: Number of fuel rods $= 61696$

$R = 8\pi k_f$: k_f is the thermal conductivity of the fuel $= 0.75398;\ W\ ℃^{-1}\ cm^{-1}$

ρ_m: Density of moderator $= 0.74;\ g\ cm^{-3}$

c_m: Specific heat capacity of moderator $= 5.6;\ J\ g^{-1}\ ℃^{-1}$

V_m: Volume of moderator $= 4.17 * 10^7;\ cm^3$

$\dot{m}$: Mass flow rate $= 4.7 * 10^6;\ g\ s^{-1}$

P_0: Initial power at equilibrium $= 2 * 10^9;\ W$

δP: Change in reactor power; W

δC: Change in precursor power; W

δT_f: Change in fuel temperature; ℃

δT_m: Change in moderator temperature; ℃

δT_{in}: Change in inlet temperature; ℃

$$\text{Let } \mathbf{S} = \begin{bmatrix} -\frac{\beta}{\tau_p} & \lambda & \frac{\beta}{\tau_p}P_0\alpha_f & \frac{\beta}{\tau_p}P_0\alpha_m \\ \frac{\beta}{\tau_p} & -\lambda & 0 & 0 \\ \frac{1}{\rho_f c_f A_r V_{cell}} & 0 & -\frac{1}{\tau_{hx}} & \frac{1}{\tau_{hx}} \\ 0 & 0 & \frac{LNR}{\rho_m c_m V_m} & \frac{-LNR - 2\dot{m}c_m}{\rho_m c_m V_m} \end{bmatrix} \text{ and } \mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{2\dot{m}c_m}{\rho_m c_m V_m} \end{bmatrix}$$

$\mathbf{F}_\mathrm{A} = \mathrm{expm}(\mathbf{S})$; where expm denotes matrix exponentiation

$\mathbf{G}_\mathrm{A} = \mathbf{S}^{-1}\,[\mathbf{I_4} - \mathrm{expm}(\mathbf{S})]\,\mathbf{T}$; where $\mathbf{I_n}$ is the n x n identity matrix

$\mathbf{H}_\mathrm{A} = [1\ 0\ 0\ 0]$

$$\mathbf{x}_\mathrm{A} = \begin{bmatrix} \delta P \\ \delta C \\ \delta T_f \\ \delta T_m \end{bmatrix}$$

Steam Generator (Model B):

h_d: Initial specific enthalpy of downcomer water $= 1.18 * 10^6$; J kg^{-1}

$\mathrm{L_d}$: Initial water level in downcomer $= 12.19$; m

x_M: Initial steam quality $= 0.268$

ρ_{pr}: Density of primary water $= 700$; kg m^{-3}

c_{pr}: Specific heat capacity of primary water $= 6.3 * 10^3$; J kg^{-1} K^{-1}

V_{pr}: Volume of primary water $= 32.5$; m^3

α_{pr}: Heat transfer coefficient from primary to metal tube $= 7 * 10^3$; W m^{-2}K^{-1}

A_{pr}: Inner area of U $-$ tubes $= 5400$; m^2

ρ_{me}: Density of U − tube metal = $8 * 10^3$; kg m^{-3}

c_{me}: Specific heat capacity of U − tube metal = 500; J kg^{-1} K^{-1}

V_{me}: Volume of U − tube metal = 6.1; m^3

α_{pr}: Heat transfer coefficient from primary to metal tube = $7 * 10^3$; W m^{-2}K^{-1}

A_{pr}: Inner area of U − tubes = 5400; m^2

α_{se}: Heat transfer coefficient from metal tube to secondary = $2 * 10^4$; W m^{-2}K^{-1}

A_{se}: Outer area of U − tubes = 6000; m^2

T_{sat}: Saturation temperature = $9.2 * 10^{-6}P + 495$; K [P in Pa]

A_3: Area of upper part of steam generator = 18.2; m^2

A_2: Area of lower part of steam generator = 9.8; m^2

A_1: Area of steam node = 8.3; m^2

L_1: Length of U − tube bundle = 10.3; m

L_2: Length of steam node = 14.8; m

L_3: Total length of steam generator = 17.1; m

L_4: Length of lower part of steam generator = 9.6; m

$$k_1 = 1 - \frac{L_1}{2L_2} = 0.65$$

$k_3 = L_4(A_2 - A_3) = -80.6$; m^3

$k_4 = A_3 - A_1 = 9.9$; m^2

V_{st}: Volume of steam dome node = 41.9; m^3

V_s: Volume of steam node = 93.1; m^3

$$W_d = k_8\left(\frac{L_d}{v_d} - (1 - x_M)\frac{A_1}{A_w}\frac{L_2}{v_s}\right)^{-\frac{1}{2}}; \text{kg s}^{-1}$$

W_{st}: Mass flow rate of steam = $\frac{\mathrm{V_{st}}}{\mathrm{v}''}$; kg s^{-1}

v^f: Specific volume of water = $3.3 * 10^{-11}Pr + 1.1 * 10^{-3}$; m^{3}kg^{-1} [$Pr$ in Pa]

v^p: Specific volume of steam = $-3.8 * 10^{-9}Pr + 5.4 * 10^{-2}$; m^{3}kg^{-1} [$Pr$ in Pa]

$v_g = v'' - v'$; m^3 kg^{-1}

v_s: Specific volume of the steam − water mixture = $v' + k_1 x_M v_g$; m^3 kg^{-1}

h^f: Specific enthalpy of saturated water = $5 * 10^{-2}Pr + 9.2 * 10^5$; J kg^{-1} [$Pr$ in Pa]

r: Heat of vaporization $= -6.3 * 10^{-2} Pr + 2 * 10^{6}$; J kg^{-1} [$Pr$ in Pa]

h_s: Specific enthalpy of the steam $-$ water mixture $= h' + k_1 x_M r$; m^3 kg^{-1}

V_d: Volume of the downcomer $= k_3 + k_4 * L_d$; m^3

v_d: Specific volume of water in downcomer $= 1.5 * 10^{-6} T + 4.7 * 10^{-4}$; m^{3}kg^{-1} [T in K]

$$x_r = \frac{1 - x_M}{x_M}$$

δT_{pr}: Change in primary temperature; K

δT_{me}: Change in metal temperature; K

δL_d: Change in steam generator water level; m

δh_d: Change in downcomer enthalpy; MJ kg^{-1}

δx_M: Change in steam quality

δPr: Change in pressure; bar

Similar to the reactor matrix exponentiation, the following matrices are computed for the above values:

$$\mathbf{F}_{\mathrm{B}} = \begin{bmatrix} 0.544888 & 0.035270 & 0.000028 & 0.000909 & 0.000075 & 0.011085 \\ 0.220770 & 0.089635 & 0.000285 & 0.009596 & 0.000783 & 0.063281 \\ 0.015234 & 0.007519 & 0.993286 & 0.001894 & -0.015324 & -0.002151 \\ 0.004929 & 0.007739 & 0.000034 & 0.967432 & -0.000369 & 0.006846 \\ 1.088189 & 0.941626 & -0.055545 & 0.139170 & 0.874916 & -0.256764 \\ 1.062617 & 0.975153 & 0.004011 & 0.136437 & 0.011042 & 0.735099 \end{bmatrix}$$

$$\mathbf{G}_{\mathrm{B}} = \begin{bmatrix} 0.314474 \\ 0.080939 \\ 0.003830 \\ 0.000717 \\ 0.233299 \\ 0.222291 \end{bmatrix}$$

$$\mathbf{H}_{\mathrm{B}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{x}_{\mathrm{B}} = \begin{bmatrix} \delta T_{pr} \\ \delta T_{me} \\ \delta L_d \\ \delta h_d \\ \delta x_M \\ \delta Pr \end{bmatrix}$$

REFERENCES

[1] J. S. Albus, "Outline for a theory of intelligence," *IEEE Trans Syst Man Cybern*, vol. 21, no. 3, pp. 473–509, May 1991, doi: 10.1109/21.97471.

[2] S. C. Kleene, "Introduction to metamathematics.," p. 550, 1971.

[3] D. E. O'Leary, "Gartner's hype cycle and information system research issues," *International Journal of Accounting Information Systems*, vol. 9, no. 4, pp. 240–252, Dec. 2008, doi: 10.1016/J.ACCINF.2008.09.001.

[4] K. Lis, M. Koryciński, and K. A. Ciecierski, "Classification of masked image data," *PLoS One*, vol. 16, no. 7, p. e0254181, Jul. 2021, doi: 10.1371/JOURNAL.PONE.0254181.

[5] K. Schawinski, C. Zhang, H. Zhang, L. Fowler, and G. Kr-ishnan Santhanam, "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit," 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, Dec. 2015, Accessed: Jul. 22, 2021. [Online]. Available: https://arxiv.org/abs/1512.03385v1

[7] L. Gondara, "Medical Image Denoising Using Convolutional Denoising Autoencoders," in *IEEE International Conference on Data Mining Workshops, ICDMW*, Jul. 2016, vol. 0, pp. 241–246. doi: 10.1109/ICDMW.2016.0041.

[8] C. Siristatidis and A. Pouliakis, "Artificial Intelligence in IVF: A Need," *Syst Biol Reprod Med*, vol. 57, no. 4, pp. 179–185, 2011, doi: 10.3109/19396368.2011.558607.

[9] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, "Systems biology informed deep learning for inferring parameters and hidden dynamics," *PLoS Comput Biol*, vol. 16, no. 11, p. e1007575, Nov. 2020, doi: 10.1371/journal.pcbi.1007575.

[10] N. Pacilio, A. Colombino, R. Mosiello, F. Norelli, and V. M. Jorio, "The Analysis of Reactor Noise: Measuring Statistical Fluctuations in Nuclear Systems," in *Advances in Nuclear Science and Technology*, 1979. doi: 10.1007/978-1-4613-2862-9_2.

[11] J. Pence, P. Farshadmanesh, J. Kim, C. Blake, and Z. Mohaghegh, "Data-theoretic approach for socio-technical risk analysis: Text mining licensee event reports of U.S. nuclear power plants," *Saf Sci*, vol. 124, p. 104574, Apr. 2020, doi: 10.1016/J.SSCI.2019.104574.

[12] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, doi: 10.1162/089976601750264965.

[13] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nat Commun*, vol. 9, no. 1, pp. 1–10, Dec. 2018, doi: 10.1038/s41467-018-07210-0.

[14] I. J. Goodfellow *et al.*, "Generative adversarial nets," 2014.

[15] J. Fletcher, "Deepfakes, Artificial Intelligence, and Some Kind of Dystopia: The New Faces of Online Post-Fact Performance," *Theatre Journal*, vol. 70, no. 4, pp. 455–471, 2018, doi: 10.1353/tj.2018.0097.

[16] B. Goertzel, "Artificial General Intelligence: Concept, State of the Art, and Future Prospects," *Journal of Artificial General Intelligence*, vol. 5, no. 1, pp. 2013–2015, 2014, doi: 10.2478/jagi-2014-0001.

[17] S. Garfinkel, J. Matthews, S. S. Shapiro, and J. M. Smith, "Toward algorithmic transparency and accountability," *Commun ACM*, vol. 60, no. 9, p. 5, Sep. 2017, doi: 10.1145/3125780.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 1135–1144, Aug. 2016, doi: 10.1145/2939672.2939778.

[19] D. Roselli, J. Matthews, and N. Talagala, "Managing Bias in AI," *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, doi: 10.1145/3308560.

[20] A. By, J. Silberg, and J. Manyika, "Notes from the AI frontier: Tackling bias in AI (and in humans)," 2019.

[21] D. Bosch, "Trump Administration Outlines Approach to Regulating AI," 2020.

[22] Human Rights Watch, "How the EU's Flawed Artificial Intelligence Regulation Endangers the Social Safety Net: Questions and Answers ," 2021. https://www.hrw.org/news/2021/11/10/how-eus-flawed-artificial-intelligence-regulation-endangers-social-safety-net#_ftn1 (accessed Nov. 26, 2021).

[23] M. R. Hassan and B. Nath, "Stock market forecasting using Hidden Markov Model: A new approach," *Proceedings - 5th International Conference on Intelligent Systems Design and Applications 2005, ISDA '05*, vol. 2005, pp. 192–196, 2005, doi: 10.1109/ISDA.2005.85.

[24] X. Wang *et al.*, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks", Accessed: Nov. 26, 2021. [Online]. Available: https://github.com/xinntao/ESRGAN.

[25] J. X. Chen, "The Evolution of Computing: AlphaGo," *Comput Sci Eng*, vol. 18, no. 4, pp. 4–7, Jul. 2016, doi: 10.1109/MCSE.2016.74.

[26] OpenAI *et al.*, "Dota 2 with Large Scale Deep Reinforcement Learning," Dec. 2019, Accessed: Nov. 25, 2021. [Online]. Available: https://arxiv.org/abs/1912.06680v1

[27] A. Elgammal, "AI Is Blurring the Definition of Artist: Advanced algorithms are using machine learning to create art autonomously," *Am Sci*, vol. 107, no. 1, pp. 18–22, Jan. 2019, Accessed: Nov. 26, 2021. [Online]. Available: https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00030996&v=2.1&it=r&id=GALE%7CA579092374&sid=googleScholar&linkaccess=fulltext

[28] Good Morning America, "Jordan Peele uses AI, President Obama in fake news PSA," *Youtube*, 2018. https://www.youtube.com/watch?v=bE1KWpoX9Hk (accessed Nov. 26, 2021).

[29] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," *Proceedings of 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2014*, 2014, doi: 10.1109/AQTR.2014.6857843.

[30] C. Beecks, S. Devasya, and R. Schlutter, "Machine Learning for Enhanced Waste Quantity Reduction: Insights from the MONSOON Industry 4.0 Project," pp. 1–6, 2019, doi: 10.1007/978-3-662-58485-9_1.

[31] Y. Li and H. S. Abdel-Khalik, "Data trustworthiness signatures for nuclear reactor dynamics simulation," *Progress in Nuclear Energy*, vol. 133, p. 103612, Mar. 2021, doi: 10.1016/j.pnucene.2020.103612.

[32] T. Papamarkou *et al.*, "Automated detection of corrosion in used nuclear fuel dry storage canisters using residual neural networks," *Nuclear Engineering and Technology*, vol. 53, no. 2, pp. 657–665, Feb. 2021, doi: 10.1016/J.NET.2020.07.020.

[33] A. Stoll, N. Pierschel, K. Wenzel, and T. Langer, "Process Control in a Press Hardening Production Line with Numerous Process Variables and Quality Criteria," *Anomaly Detection in Manufacturing*, pp. 77–86, 2019, doi: 10.1007/978-3-662-58485-9_9.

[34] A. Kuhnle and G. Lanza, “Application of Reinforcement Learning in Production Planning and Control of Cyber Physical Production Systems,” *Anomaly Detection in Manufacturing*, pp. 123–132, 2019, doi: 10.1007/978-3-662-58485-9_14.

[35] A. Graß, C. Beecks, and J. A. C. Soto, “Unsupervised Anomaly Detection in Production Lines,” *Anomaly Detection in Manufacturing*, pp. 18–25, 2019, doi: 10.1007/978-3-662-58485-9_3.

[36] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, “Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data,” *IEEE Trans Industr Inform*, 2019, doi: 10.1109/TII.2019.2891261.

[37] A. Sundaram, H. S. Abdel-Khalik, and A. al Rashdan, “Deceptive Infusion of Data (DIOD) for Nuclear Reactors,” *Proceedings of M&C2021*, 2021.

[38] Y. Li, H. S. Abdel-Khalik, A. J. Brunett, E. Jennings, T. Mui, and R. Hu, “ROM-Based Surrogate Systems Modeling of EBR-II,” *https://doi.org/10.1080/00295639.2020.1840238*, vol. 195, no. 5, pp. 520–537, 2020, doi: 10.1080/00295639.2020.1840238.

[39] Y. Li, E. Bertino, and H. S. Abdel-Khalik, “Effectiveness of Model-Based Defenses for Digitally Controlled Industrial Systems: Nuclear Reactor Case Study,” *Nucl Technol*, 2020, doi: 10.1080/00295450.2019.1626170.

[40] C. Rabiti *et al.*, “System Reliability Analysis Capability and Surrogate Model Application in RAVEN,” Nov. 2015, doi: 10.2172/1376079.

[41] D. Huang and H. S. Abdel-Khalik, “Theoretical Development of Cross Section Uncertainty Library for Core Simulators,” *Journal of Nuclear Engineering and Radiation Science*, vol. 6, no. 1, Jan. 2020, doi: 10.1115/1.4045031.

[42] D. Huang and H. S. Abdel-Khalik, “Development of Uncertainty Quantification Capability for NESTLE,” *International Conference on Nuclear Engineering, Proceedings, ICONE*, vol. 9, Oct. 2017, doi: 10.1115/ICONE25-67797.

[43] D. Huang and H. S. Abdel-Khalik, “Application of Cross Sections Uncertainty Propagation Framework to Light and Heavy Water Reactor Systems,” *Journal of Nuclear Engineering and Radiation Science*, vol. 6, no. 1, Jan. 2020, doi: 10.1115/1.4045032.

[44] D. Huang, H. Abdel-Khalik, C. Rabiti, and F. Gleicher, “Dimensionality reducibility for multi-physics reduced order modeling,” *Ann Nucl Energy*, vol. 110, pp. 526–540, Dec. 2017, doi: 10.1016/J.ANUCENE.2017.06.045.

[45] Y. Bang, H. S. Abdel-Khalik, and J. M. Hite, “Hybrid reduced order modeling applied to nonlinear models,” *Int J Numer Methods Eng*, 2012, doi: 10.1002/nme.4298.

[46] Y. Bang and H. S. Abdel-Khalik, “Reduced Order Modeling for Multi-Physics Problems,” *Trans Am Nucl Soc*, vol. 107, pp. 586–588, 2012, doi: 10.1002/nme.4298.

[47] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, and X. Bellekens, “Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection,” *Electronics 2020, Vol. 9, Page 1684*, vol. 9, no. 10, p. 1684, Oct. 2020, doi: 10.3390/ELECTRONICS9101684.

[48] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, “Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices,” *IEEE Internet Things J*, 2021, doi: 10.1109/JIOT.2021.3100755.

[49] C. M. Ahmed, J. Zhou, and A. P. Mathur, “Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate sensors in CPS,” *34Th Annual Computer Security Applications Conference (ACSAC 2018)*, 2018, doi: 10.1145/3274694.3274748.

[50] D. I. Urbina *et al.*, “Limiting the Impact of Stealthy Attacks on Industrial Control Systems,” 2016, doi: 10.1145/2976749.2978388.

[51] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Data-driven discovery of partial differential equations,” *Sci Adv*, vol. 3, no. 4, p. e1602614, Apr. 2017, doi: 10.1126/sciadv.1602614.

[52] S. L. Brunton, J. L. Proctor, J. N. Kutz, and W. Bialek, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proc Natl Acad Sci U S A*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016, doi: 10.1073/pnas.1517384113.

[53] H. H. Tan and K. H. Lim, “Vanishing Gradient Mitigation with Deep Learning Neural Network Optimization,” *2019 7th International Conference on Smart Computing and Communications, ICSCC 2019*, Jun. 2019, doi: 10.1109/ICSCC.2019.8843652.

[54] N. Tishby and N. Zaslavsky, “Deep Learning and the Information Bottleneck Principle,” *2015 IEEE Information Theory Workshop, ITW 2015*, Mar. 2015, Accessed: Jun. 13, 2021. [Online]. Available: http://arxiv.org/abs/1503.02406

[55] J.-P. Vert, K. Tsuda, and B. Schölkopf, “A primer on kernel methods.”

[56] N. ben Amor, S. Benferhat, and Z. Elouedi, “Naive Bayes vs decision trees in intrusion detection systems,” in *Proceedings of the ACM Symposium on Applied Computing*, 2004, vol. 1, pp. 420–424. doi: 10.1145/967900.967989.

[57] D. Md. Farid, N. Harbi, and M. Z. Rahman, “Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection,” *International journal of Network Security & Its Applications*, vol. 2, no. 2, pp. 12–25, May 2010, doi: 10.5121/ijnsa.2010.2202.

[58] S. A. Dudani, “The Distance-Weighted k-Nearest-Neighbor Rule,” *IEEE Trans Syst Man Cybern*, vol. SMC-6, no. 4, pp. 325–327, 1976, doi: 10.1109/TSMC.1976.5408784.

[59] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015.

[60] A. Sundaram, H. S. Abdel-Khalik, and O. Ashy, “A data analytical approach for assessing the efficacy of Operational Technology active defenses against insider threats,” *Progress in Nuclear Energy*, vol. 124, p. 103339, Jun. 2020, doi: 10.1016/j.pnucene.2020.103339.

[61] Y. Mo, S. Weerakkody, and B. Sinopoli, “Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs,” *IEEE Control Syst*, 2015, doi: 10.1109/MCS.2014.2364724.

[62] “The Colors of Noise - Colors of noise - Wikipedia.” https://en.wikipedia.org/wiki/Colors_of_noise#/media/File:The_Colors_of_Noise.png (accessed Nov. 28, 2021).

[63] G. S. Vernam, “Cipher printing telegraph systems: For secret wire and radio telegraphic communications,” *Journal of the A.I.E.E.*, vol. 45, no. 2, pp. 109–115, Jul. 2013, doi: 10.1109/JAIEE.1926.6534724.

[64] A. Sundaram and H. Abdel-Khalik, “Covert Cognizance: A Novel Predictive Modeling Paradigm,” *Nucl Technol*, 2021, doi: 10.1080/00295450.2020.1812349.

[65] A. Sundaram and H. Abdel-Khalik, “Validation of Covert Cognizance Active Defenses,” *Nuclear Science and Engineering*, 2021, doi: 10.1080/00295639.2021.1897731.

[66] D. R. Farley, M. G. Negus, and R. N. Slaybaugh, “Industrial Internet-of-Things & Data Analytics for Nuclear Power & Safeguards.,” Nov. 2018, doi: 10.2172/1481947.

[67] N. A. Qarabsh, S. S. Sabry, and H. A. Qarabash, “Smart grid in the context of industry 4.0: An overview of communications technologies and challenges,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 656–665, 2020, doi: 10.11591/ijeecs.v18.i2.pp656-665.

[68] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” *IEEE Secur Priv*, vol. 9, no. 3, pp. 49–51, May 2011, doi: 10.1109/MSP.2011.67.

[69] NIST, “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1,” Gaithersburg, MD, Apr. 2018. doi: 10.6028/NIST.CSWP.04162018.

[70] N. Sayfayn and S. Madnick, “Cybersafety Analysis of the Maroochy Shire Sewage Spill Cybersafety Analysis of the Maroochy Shire Sewage Spill (Preliminary Draft),” 2017.

[71] T. J. Williams, “The Purdue enterprise reference architecture,” *Comput Ind*, vol. 24, no. 2–3, pp. 141–158, Sep. 1994, doi: 10.1016/0166-3615(94)90017-5.

[72] T. J. Williams, “The Purdue Enterprise Reference Architecture,” *IFAC Proceedings Volumes*, vol. 26, no. 2, pp. 559–564, Jul. 1993, doi: 10.1016/S1474-6670(17)48532-6.

[73] “ISA 95.” https://isa-95.com/ (accessed Nov. 19, 2021).

[74] P. Ackerman, “The Purdue model for Industrial control systems,” *Industrial Cybersecurity*, 2017, Accessed: Nov. 28, 2021. [Online]. Available: https://www.packtpub.com/mapt/book/networking_and_servers/9781788395151/1/ch01lvl1sec10/the-purdue-model-for-industrial-control-systems

[75] A. H. Sayed and C. G. Lopes, “Distributed processing over adaptive networks,” *2007 9th International Symposium on Signal Processing and its Applications, ISSPA 2007, Proceedings*, 2007, doi: 10.1109/ISSPA.2007.4555636.

[76] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[77] Shmueli Erez, Vaisenberg Ronen, Elovici Yuval, and Glezer Chanan, “Database encryption,” *ACM SIGMOD Record*, vol. 38, no. 3, pp. 29–34, Dec. 2010, doi: 10.1145/1815933.1815940.

[78] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: A survey,” *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018, doi: 10.1504/IJWGS.2018.095647.

[79] S. Cheng, B. Zeng, and Y. Z. Huang, "Research on application model of blockchain technology in distributed electricity market," *IOP Conf Ser Earth Environ Sci*, vol. 93, no. 1, p. 012065, Nov. 2017, doi: 10.1088/1755-1315/93/1/012065.

[80] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*, Nov. 2016, doi: 10.1109/HEALTHCOM.2016.7749510.

[81] T. M. Choi, "Blockchain-technology-supported platforms for diamond authentication and certification in luxury supply chains," *Transp Res E Logist Transp Rev*, vol. 128, pp. 17–29, Aug. 2019, doi: 10.1016/J.TRE.2019.05.011.

[82] M. Eckhart and A. Ekelhart, "Digital Twins for Cyber-Physical Systems Security: State of the Art and Outlook," in *Security and Quality in Cyber-Physical Systems Engineering*, 2019. doi: 10.1007/978-3-030-25312-7_14.

[83] Y. Mo and B. Sinopoli, "Secure control against replay attacks," *2009 47th Annual Allerton Conf on CCC*, 2009. doi: 10.1109/ALLERTON.2009.5394956.

[84] Y. Yuan and Y. Mo, "Security in cyber-physical systems: Controller design against Known-Plaintext Attack," *2015 54th IEEE CDC*, 2015. doi: 10.1109/CDC.2015.7403133.

[85] Y. Ni, Z. Guo, Y. Mo, and L. Shi, "On the Performance Analysis of Reset Attack in Cyber-Physical Systems," *IEEE Trans Automat Contr*, 2020, doi: 10.1109/TAC.2019.2914655.

[86] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 1-33, 2011. doi: 10.1145/1952982.1952995.

[87] F. Hou and J. Sun, "False data injection attacks in cyber-physical systems based on inaccurate model," *IECON 2017*, 2017. doi: 10.1109/IECON.2017.8217004.

[88] J. Toon, "Baking and Boiling Botnets Could Drive Energy Market Swings and Damage," *Georgia Tech News Center*, 2020. https://news.gatech.edu/2020/08/04/baking-and-boiling-botnets-could-drive-energy-market-swings-and-damage (accessed Aug. 11, 2020).

[89] C. Barreto, A. A. Cárdenas, N. Quijano, and E. Mojica-Nava, "CPS: Market analysis of attacks against demand response in the smart grid," *ACM International Conference Proceeding Series*, vol. 2014-December, no. December, pp. 136–145, Dec. 2014, doi: 10.1145/2664243.2664284.

[90] H. Fawzi, P. Tabuada, and S. Diggavi, “Secure estimation and control for cyber-physical systems under adversarial attacks,” *IEEE Trans Automat Contr*, 2014, doi: 10.1109/TAC.2014.2303233.

[91] R. Zhang and P. Venkitasubramaniam, “Stealthy control signal attacks in vector LQG systems,” 2016 *American Control Conf*, 2016. doi: 10.1109/ACC.2016.7525077.

[92] J. D. Gilsinn and R. Schierholz, “Security Assurance Levels: A Vector Approach to Describing Security Requirements”, *NIST*, 2010.

[93] J. Larsen, “Miniaturization,” *Las Vegas Black Hat 2014*, 2014.

[94] T. Tsvetanov and S. Slaria, “The effect of the Colonial Pipeline shutdown on gasoline prices,” *Econ Lett*, vol. 209, p. 110122, Dec. 2021, doi: 10.1016/J.ECONLET.2021.110122.

[95] A. Sundaram and H. S. Abdel-Khalik, “Developing Covert Cognizance (C2) for Industrial Control Systems,” *Proceedings of M&*C2021, 2021.

[96] M. H. Almeshekah and E. H. Spafford, “Cyber security deception,” in *Cyber Deception: Building the Scientific Foundation*, Springer International Publishing, 2016, pp. 23–50. doi: 10.1007/978-3-319-32699-3_2.

[97] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering, Transactions of the ASME*, 1960, doi: 10.1115/1.3662552.

[98] B. D. Brumback and M. D. Srinath, “CHI-SQUARE TEST FOR FAULT-DETECTION IN KALMAN FILTERS.,” *IEEE Trans Automat Contr*, vol. AC-32, no. 6, pp. 552–554, 1987, doi: 10.1109/tac.1987.1104658.

[99] D. Li, D. Chen, J. Goh, and S.-K. Ng, “Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series.” Accessed: Oct. 16, 2020. [Online]. Available: https://github.com/LiDan456/GAN-AD

[100] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, “MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks.” Accessed: Oct. 16, 2020. [Online]. Available: https://github.com/LiDan456/MAD-GANs

[101] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[102] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, “Long Short Term Memory Networks for Anomaly Detection in Time Series,” *ESANN 2015*, 2015.

[103] E. Gamal, “What is long short-term memory (LSTM)?,” *Quora*, Jun. 2020. https://www.quora.com/What-is-long-short-term-memory-LSTM (accessed Nov. 28, 2021).

[104] S. Risi and M. Preuss, “Behind DeepMind’s AlphaStar AI that Reached Grandmaster Level in StarCraft II,” *KI - Künstliche Intelligenz 2020 34:1*, vol. 34, no. 1, pp. 85–86, Feb. 2020, doi: 10.1007/S13218-020-00642-1.

[105] D. Bruneo and F. de Vita, “On the use of LSTM networks for predictive maintenance in smart industries,” *Proceedings - 2019 IEEE International Conference on Smart Computing, SMARTCOMP 2019*, pp. 241–248, Jun. 2019, doi: 10.1109/SMARTCOMP.2019.00059.

[106] N. Tax, I. Verenich, M. la Rosa, and M. Dumas, “Predictive Business Process Monitoring with LSTM Neural Networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10253 LNCS, pp. 477–492, Dec. 2016, doi: 10.1007/978-3-319-59536-8_30.

[107] T. Silva, “A Short Introduction to Generative Adversarial Networks.” https://sthalles.github.io/intro-to-gans/ (accessed Nov. 28, 2021).

[108] A. Zhavoronkov *et al.*, “Deep learning enables rapid identification of potent DDR1 kinase inhibitors,” *Nature Biotechnology 2019 37:9*, vol. 37, no. 9, pp. 1038–1040, Sep. 2019, doi: 10.1038/s41587-019-0224-x.

[109] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation forest,” *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 413–422, 2008, doi: 10.1109/ICDM.2008.17.

[110] H. Elmqvist, “DYMOLA - A STRUCTURED MODEL LANGUAGE FOR LARGE CONTINUOUS SYSTEMS.,” *Proceedings of the Summer Computer Simulation Conference*, 1980.

[111] M. Dempsey, “Dymola for multi-engineering modelling and simulation,” 2006. doi: 10.1109/VPPC.2006.364294.

[112] Y. Zhao, Z. Nasrullah, and Z. Li, “PyOD: A Python Toolbox for Scalable Outlier Detection,” *Journal of Machine Learning Research*, vol. 20, pp. 1–7, 2019, Accessed: Sep. 05, 2022. [Online]. Available: https://pyod.readthedocs.io

[113] M. S. Greenwood, B. R. Betzler, A. lou Qualls, J. Yoo, and C. Rabiti, “Demonstration of the Advanced Dynamic System Modeling Tool TRANSFORM in a Molten Salt Reactor Application via a Model of the Molten Salt Demonstration Reactor,” *Nucl Technol*, vol. 206, no. 3, pp. 478–504, Mar. 2020, doi: 10.1080/00295450.2019.1627124.

[114] A. Sundaram, H. S. Abdel-Khalik, D. Roberson, and M. el Hariri, “Data recovery via covert cognizance for unattended operational resilience,” *Progress in Nuclear Energy*, vol. 151, p. 104317, Sep. 2022, doi: 10.1016/J.PNUCENE.2022.104317.

[115] S. Kirshenberg, H. Jackler, J. Eun, K. Rock, B. Oakley, and W. Goldenberg, “Small Modular Reactors: Adding to Resilience at Federal Facilities SMALL MODULAR REACTORS: ADDING TO RESILIENCE AT FEDERAL FACILITIES CONTENTS,” 2017.

[116] V. Agarwal, Y. A. Ballout, and J. C. Gehin, “Fission Battery Initiative,” 2021.

[117] D. Program Manager and J. H. Jackson, “How will MARVEL complement MAGNET?”.

[118] “A Microreactor Program Plan for The Department of Energy An Integrated, Strategic Program Plan for Research and Development supporting Demonstration and Deployment of Nuclear Microreactors,” 2021.

[119] K. le Blanc *et al.*, “A human automation interaction concept for a small modular reactor control room - 101.” 2017.

[120] D. E. Holcomb and R. T. Wood, “U.S. Department Of Energy Advanced Small Modular Reactor R&D Program: Instrumentation, Controls, and Human-Machine Interface,” May 2013.

[121] D. A. Clayton and R. T. Wood, “The Role of Instrumentation and Controls Technology in Enabling Deployment of Small Modular Reactors,” 2010.

[122] R. T. Wood, B. R. Upadhyaya, and D. C. Floyd, “An autonomous control framework for advanced reactors,” *Nuclear Engineering and Technology*, vol. 49, no. 5, pp. 896–904, Aug. 2017, doi: 10.1016/J.NET.2017.07.001.

[123] S. M. Cetiner *et al.*, “Supervisory Control System for Multi-Modular Advanced Reactors,” 2016.

[124] S. M. Cetiner, M. D. Muhlheim, G. F. Flanagan, D. L. Fugate, and R. A. Kisner, “Development of an Automated Decision-Making Tool for Supervisory Control System,” ORNL/TM-2014/363, Sep. 2014, doi: 10.2172/1252136.

[125] P. Ramuhalli and S. M. Cetiner, “Concepts for Autonomous Operation of Microreactors,” 2019.

[126] Belle R. Upadhyaya, K. Zhao, S.R.P. Perillo, Xiaojia Xu, and M.G. Na, "Autonomous Control of Space Reactor Systems," DE-FG07-04ID14589/UTNE-06, Nov. 2007, doi: 10.2172/920996.

[127] H. R. Trellue, J. O'Brien, J. S. Carpenter, R. S. Reid, D. Guillen, and P. Sabharwall, "Microreactor Demonstration and Testing Progress in FY19," LA-UR-19-28768, Sep. 2019.

[128] D. Hadziosmanovic, R. Sommer, E. Zambon, and P. H. Hartel, "Through the Eye of the PLC: Semantic Security Monitoring for Industrial Processes", *ICPS Proceedings ACSAC '14*, 2014, doi: 10.1145/2664243.2664277.

[129] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Semantic Security Analysis of SCADA Networks to Detect Malicious Control Commands in Power Grids," *Proceedings of the first ACM workshop on Smart energy grid security - SEGS '13*, 2013, doi: 10.1145/2516930.

[130] D. Myers, K. Radke, S. Suriadi, and E. Foo, "Process Discovery for Industrial Control System Cyber Attack Detection," *IFIP Adv Inf Commun Technol*, vol. 502, pp. 61–75, 2017, doi: 10.1007/978-3-319-58469-0_5.

[131] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, "State-Based Network Intrusion Detection Systems for SCADA Protocols: A Proof of Concept," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6027 LNCS, pp. 138–150, 2009, doi: 10.1007/978-3-642-14379-3_12.

[132] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Research Article International Journal of Distributed Sensor Networks*, vol. 14, no. 8, 2018, doi: 10.1177/1550147718794615.

[133] D. Roberson and J. F. O'Brien, "Variable loop gain using excessive regeneration detection for a delayed wide-area control system," *IEEE Trans Smart Grid*, vol. 9, no. 6, pp. 6623–6632, Nov. 2018, doi: 10.1109/TSG.2017.2717449.

[134] A. O. de Sá, A. Casimiro, R. C. S. Machado, and L. F. R. da C. Carmo, "Identification of data injection attacks in networked control systems using noise impulse integration," *Sensors (Switzerland)*, vol. 20, no. 3, pp. 4–6, Feb. 2020, doi: 10.3390/s20030792.

[135] R. Venkatakrishnan and M. A. Vouk, "The Internet of Things Using Redundancy to Detect Security Anomalies: Towards IoT security attack detectors Ubiquity Symposium", Accessed: Sep. 28, 2021. [Online]. Available: http://ubiquity.acm.org1

[136] Dell, “Recovering from a Destructive Cyber-attack,” 2017. Accessed: Sep. 28, 2021. [Online]. Available: https://www.emc.com/collateral/whitepaper/recovering-business-destructive-cyber-attack.pdf

[137] T. A. Severson, B. Croteau, E. J. Rodríguez-Seda, K. Kiriakidis, R. Robucci, and C. Patel, “A resilient framework for sensor-based attacks on cyber-physical systems using trust-based consensus and self-triggered control,” *Control Eng Pract*, vol. 101, p. 104509, 2020, doi: 10.1016/j.conengprac.2020.104509.

[138] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. MIT Press, 1992.

[139] G. S. Vernam, “Cipher Printing Telegraph Systems: For Secret Wire and Radio Telegraphic Communications,” *Transactions of the American Institute of Electrical Engineers*, 1926, doi: 10.1109/T-AIEE.1926.5061224.

[140] A. Sundaram, H. Abdel-Khalik, and A. al Rashdan, “Deceptive Infusion of Data: A Novel Data Masking Paradigm for High-Valued Systems,” *Nucl Sci and Eng*, pp. 1–16, Apr. 2022, doi: 10.1080/00295639.2022.2043542.

[141] A. Sundaram, H. S. Abdel-Khalik, and M. G. Abdo, “Preventing Reverse Engineering of Critical Industrial Data with DIOD,” *Nucl Technol*, 2022, doi: 10.1080/00295450.2022.2102848.

[142] P. Borasi, S. Khan, and V. Kumar, “Big Data and Business Analytics Market,” 2021. Accessed: Nov. 27, 2021. [Online]. Available: https://www.alliedmarketresearch.com/big-data-and-business-analytics-market

[143] H. P. Luhn, “A Business Intelligence System,” *IBM J Res Dev*, vol. 2, no. 4, pp. 314–319, Apr. 2010, doi: 10.1147/rd.24.0314.

[144] H. J. Watson and B. H. Wixom, “The current state of business intelligence,” *Computer (Long Beach Calif)*, vol. 40, no. 9, pp. 96–99, Sep. 2007, doi: 10.1109/MC.2007.331.

[145] O. Abbosh, K. Bissell, and Accenture, “Securing the Digital Economy: Reinventing the Internet | Accenture,” 2019. Accessed: Jan. 26, 2021. [Online]. Available: https://www.accenture.com/us-en/insights/cybersecurity/_acnmedia/Thought-Leadership-Assets/PDF/Accenture-Securing-the-Digital-Economy-Reinventing-the-Internet-for-Trust.pdf#zoom=50

[146] T. Tung, D. Treat, J.-L. Chatelain, P. Connolly, and Accenture, “Maximizing Collaboration Through Secure Data Sharing | Accenture,” 2019.

[147] J. Willard, X. Jia, M. Steinbach, V. Kumar, and S. Xu, "Integrating Physics-Based Modeling With Machine Learning: A Survey," vol. 1, p. 34, 2020, doi: 10.1145/1122445.1122456.

[148] S. Arridge, P. Maass, O. Öktem, and C. B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, May 2019, doi: 10.1017/S0962492919000059.

[149] J. Morshed and J. J. Kaluarachchi, "Parameter estimation using artificial neural network and genetic algorithm for free-product migration and recovery," *Water Resour Res*, vol. 34, no. 5, pp. 1101–1113, May 1998, doi: 10.1029/98WR00006.

[150] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, "NETT: Solving inverse problems with deep neural networks," *Inverse Probl*, vol. 36, no. 6, p. 065005, Jun. 2020, doi: 10.1088/1361-6420/ab6d57.

[151] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Probl*, vol. 33, no. 12, Nov. 2017, doi: 10.1088/1361-6420/aa9581.

[152] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep Learning Techniques for Inverse Problems in Imaging," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[153] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J Comput Phys*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.

[154] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *J Comput Phys*, vol. 357, pp. 125–141, Mar. 2018, doi: 10.1016/j.jcp.2017.11.039.

[155] A. F. Villaverde and J. R. Banga, "Reverse engineering and identification in systems biology: Strategies, perspectives and challenges," *J R Soc Interface*, vol. 11, no. 91, Feb. 2014, doi: 10.1098/rsif.2013.0505.

[156] M. Grimaldi, R. Visintainer, and G. Jurman, "Regnann: Reverse engineering gene networks using artificial neural networks," *PLoS One*, vol. 6, no. 12, p. 28646, Dec. 2011, doi: 10.1371/journal.pone.0028646.

[157] M. Genzel, J. Macdonald, and M. M. ¨ Arz, "Solving Inverse Problems With Deep Neural Networks-Robustness Included?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Nov. 2020.

[158] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proc Natl Acad Sci U S A*, vol. 104, no. 24, pp. 9943–9948, Jun. 2007, doi: 10.1073/pnas.0609476104.

[159] K. Yanamandra, G. L. Chen, X. Xu, G. Mac, and N. Gupta, "Reverse engineering of additive manufactured composite part by toolpath reconstruction using imaging and machine learning," *Compos Sci Technol*, vol. 198, p. 108318, Sep. 2020, doi: 10.1016/j.compscitech.2020.108318.

[160] E. C. R. Shin, D. Song, and R. Moazzezi, "Recognizing Functions in Binaries with Neural Networks," *Proc. 24th USENIX Conf on Sec Symposium*, pp. 611-626, Aug. 2015.

[161] A. L. Boone, I. v. Floros, and S. A. Johnson, "Redacting proprietary information at the initial public offering," *J financ econ*, vol. 120, no. 1, pp. 102–123, Apr. 2016, doi: 10.1016/j.jfineco.2015.06.016.

[162] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*, Springer Berlin Heidelberg, 2008, pp. 1–19. doi: 10.1007/978-3-540-79228-4_1.

[163] C. Dwork, A. Roth, C. Dwork, and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends R in Theoretical Computer Science*, vol. 9, pp. 211–407, 2014, doi: 10.1561/0400000042.

[164] P. Barbosa, A. Brito, and H. Almeida, "A Technique to provide differential privacy for appliance usage in smart metering," *Inf Sci (N Y)*, vol. 370–371, pp. 355–367, Nov. 2016, doi: 10.1016/j.ins.2016.08.011.

[165] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," *2007 IEEE 23rd Intl Conf on Data Engineering*, 2007.

[166] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-Diversity: Privacy beyond k-anonymity," in *Proceedings - International Conference on Data Engineering*, 2006, vol. 2006, p. 24. doi: 10.1109/ICDE.2006.1.

[167] A. Oppermann, F. Grasso-Toro, A. Yurchenko, and J. P. Seifert, “Secure cloud computing: Communication protocol for multithreaded fully homomorphic encryption for remote data processing,” in *Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017*, May 2018, pp. 503–510. doi: 10.1109/ISPA/IUCC.2017.00084.

[168] “California Consumer Privacy Act (CCPA) | State of California - Department of Justice - Office of the Attorney General.” https://oag.ca.gov/privacy/ccpa (accessed Jul. 21, 2021).

[169] “General Data Protection Regulation (GDPR) – Official Legal Text.” https://gdpr-info.eu/ (accessed Jul. 21, 2021).

[170] N. Carrara and J. Ernst, “On the Estimation of Mutual Information,” *Proc West Mark Ed Assoc Conf*, vol. 33, no. 1, p. 31, Jan. 2020, doi: 10.3390/proceedings2019033031.

[171] M. I. Belghazi *et al.*, “Mutual Information Neural Estimation,” *Proc MLR2018*, 2018.

[172] A. Kraskov, H. Stoegbauer, and P. Grassberger, “Estimating Mutual Information,” *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, vol. 69, no. 6, p. 16, May 2003, doi: 10.1103/physreve.69.066138.

[173] J. B. Kinney and G. S. Atwal, “Equitability, mutual information, and the maximal information coefficient,” *Proc Natl Acad Sci U S A*, vol. 111, no. 9, pp. 3354–3359, Mar. 2014, doi: 10.1073/PNAS.1309933111.

[174] S. Meyen, “Relation between classification accuracy and mutual information in equally weighted classification tasks,” *Computer Science*, 2016.

[175] M. Gavish and D. L. Donoho, “The optimal hard threshold for singular values is 4/√3,” *IEEE Trans Inf Theory*, vol. 60, no. 8, pp. 5040–5053, 2014, doi: 10.1109/TIT.2014.2323359.

[176] R. Vautard, P. Yiou, and M. Ghil, “Singular-spectrum analysis: A toolkit for short, noisy chaotic signals,” *Physica D*, vol. 58, no. 1–4, pp. 95–126, Sep. 1992, doi: 10.1016/0167-2789(92)90103-T.

[177] N. Golyandina and D. Stepanov, “SSA-based approaches to analysis and forecast of multidimensional time series.” *Computer Science*, 2012.

[178] F. Rosenblatt, “THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1,” *Psychol Rev*, vol. 65, no. 6, pp. 19–27, 1958.

[179] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “Training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 1992, pp. 144–152. doi: 10.1145/130385.130401.

[180] B. C. Moore, “Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction,” *IEEE Trans Automat Contr*, 1981, doi: 10.1109/TAC.1981.1102568.

[181] B. Schölkopf, A. Smola, and K. R. Müller, “Kernel principal component analysis,” 1997. doi: 10.1007/bfb0020217.

[182] D. Charte, F. Charte, M. J. del Jesus, and F. Herrera, “An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges $”

[183] S. E. Otto and C. W. Rowley, “Linearly-Recurrent Autoencoder Networks for Learning Dynamics,” *SIAM J Appl Dyn Syst*, vol. 18, no. 1, pp. 558–593, Dec. 2017.

PUBLICATIONS

Journal Publications

A. Sundaram, H.S. Abdel-Khalik, and M. Abdo, "Preventing Reverse-Engineering of Critical Industrial Data with DIOD," *Nuclear Technology*, 2022

A. Sundaram, H.S. Abdel-Khalik, D. Roberson, and M. El Hariri, "Data Recovery via Covert Cognizance for Unattended Operational Resilience," *Progress in Nuclear Energy*, 2022

A. Sundaram, H.S. Abdel-Khalik, and A. Al Rashdan, "Deceptive Infusion of Data (DIOD): A Novel Data Masking Paradigm for High-Valued Systems," *Nuclear Science & Engineering*, 2022

A. Sundaram, Y. Li, and H.S. Abdel-Khalik, "Denoising Algorithm for Subtle Anomaly Detection," *Nuclear Technology*, 2022

A. Sundaram, and H.S. Abdel-Khalik, "Validation of Covert Cognizance Active Defenses," *Nuclear Science & Engineering*, 2021

A. Sundaram, and H.S. Abdel-Khalik, "Covert Cognizance: A Novel Predictive Modeling Paradigm," *Nuclear Technology*, 2021

A. Sundaram, H.S. Abdel-Khalik, and O. Ashy, "A data analytical approach for assessing the efficacy of operational technology active defenses against insider threats," *Progress in Nuclear Energy*, 2020

Y. Li, A. Sundaram, H.S. Abdel-Khalik, and P.W. Talbot, "Real-time Monitoring for Detection of Adversarial Subtle Process Variations," *Nuclear Science & Engineering*, 2022

T. Lewis, A. Sundaram, H.S. Abdel-Khalik, P.W. Talbot, and C. Rabiti, "Entropy Criterion for Surrogate Timeseries Data Generation via Non-Parametric Dimensionality Reduction," *Annals of Nuclear Energy* (under review)

Conference Proceedings

T. Lewis, A. Sundaram, and H.S. Abdel-Khalik, "Extending DIOD to Regression," *Transactions of the American Nuclear Society Winter Meeting 2022*, 2022

A. Sundaram, T. Lewis, and H.S. Abdel-Khalik, "Image Synthesis of Nuclear Data using Deceptive Infusion of Data," *Transactions of the American Nuclear Society Annual Meeting 2022*, 2022

A. Sundaram, H.S. Abdel-Khalik, and A. Al Rashdan, "Deceptive Infusion of Data (DIOD) for Nuclear Reactors," *Transactions of the American Nuclear Society Winter Meeting 2021*, 2021

A. Sundaram, and H.S. Abdel-Khalik, "Developing Covert Cognizance (C2) for Industrial Control Systems," *Proceedings of M&C2021*, 2021

A. Sundaram, Y. Li, and H.S. Abdel-Khalik, "A multi-level feature extraction and denoising approach to detect subtle variations in industrial control systems," *Proceedings of M&C2021*, 2021

A. Sundaram, H.S. Abdel-Khalik, and O. Ashy, "Exploratory Study into the Effectiveness of Active Monitoring Techniques," *Transactions of the American Nuclear Society Winter Meeting 2019*, 2019