PASSIVE METHODS FOR DETECTION OF SUBTLE PROCESS VARIATIONS

by

Yeni LI

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

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

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

Dr. Elisa Bertino
Department of Computer Science

Dr. Robert S. Bean
School of Nuclear Engineering

Dr. Martin Lopez-De-Bertodano
School of Nuclear Engineering

Dr. Stylianos Chatzidakis
School of Nuclear Engineering

Approved by:
Dr. Seungjin Kim

To mom and dad, for endless love and unconditional trust

To Haozhi, for encouragement and support as always

To grandpa, for earliest inspiration and care

To grandma, in loving memories

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my advisor, Dr. Abdel-Khalik, for your patience, heuristic education, profound erudition, and continuous support throughout my Ph.D. study. Your guidance helped me during all the research and writing of this thesis. Dr. Abdel-Khalik, who never sleeps, is one of the most energetic and ingenious people I have ever known. I hope I could be as helpful, lively, enthusiastic, and responsible as you. I could not have imagined a better mentor for my Ph.D. study.

I would also like to extend my gratitude to my committee members, Dr. Bertodano, Dr. Bean, Dr. Bertino, and Dr. Chatzidakis. Dr. Paul W. Talbot was invited to attend both my preliminary and final examinations. Your invaluable advice and insightful questions completed this dissertation. I will continue to improve my skills and pass along the lessons I learned.

My words will fail to express my heartfelt thanks to my colleagues and friends, Dongli Huang, Jia Zhou, Arvind Sundaram, Zhuoran Dang, Gang Yang, Haoxuan Wang, Jeongwon Seo, and many other people who have provided countless help, joy, and great company during this journey.

I also owe much gratitude to my family. My caring mom, Yan Yan, and dad, Yinghua Li, have provided unconditional love, unwavering support, and unswerving faith in me. I would not have made it this far without you. Special thanks to my fiancé, Haozhi, who has been non-judgemental towards me and instrumental in letting out my negativity and anxiety. Your persistence solidifies your commitment and renews my faith in our long-distance relationship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

As industries take advantage of the widely adopted digitalization of industrial control systems, concerns are heightened about their potential vulnerability to adversarial attacks. False data injection attack is one of the most realistic threats because the attack could be as simple as performing a reply attack allowing attackers to circumvent conventional anomaly detection methods. This attack scenario is real for critical systems, e.g., nuclear reactors, chemical plants, etc., because physics-based simulators for a wide range of critical systems can be found in the open market providing the means to generate physics-conforming attack. The state-of-the-art monitoring techniques have proven effective in detecting sudden variations from established recurring patterns, derived by model-based or data-driven techniques, considered to represent normal behavior. This Ph. D. work further develops a new method designed to detect subtle variations expected with stealthy attacks that rely on intimate knowledge of the system. The method employs physics modeling and feature engineering to design mathematical features that can detect subtle deviations from normal process variation. This work extends the method to real-time analysis and employs a new denoising filter to ensure resiliency to noise, i.e., ability to distinguish subtle variations from normal process noise. The method applicability is exemplified using a hypothesized triangle attack, recently demonstrated to be extremely effective in bypassing detection by conventional monitoring techniques, applied to a representative nuclear reactor system model using the RELAP5 computer code.

1 INTRODUCTION

1.1 Overview

Due to the wide range of advantages resulting from digitization, most critical infrastructure systems, such as nuclear, chemical, oil and gas plants, water treatment facilities, refineries, etc., are resorting to full digitization strategies for the control systems used to regulate their industrial processes. To an attacker, digitization offers a whole new realm of possibilities to compromise and then commandeer the operation of critical systems, considered to be the first target of a state-sponsored crippling attack against a country. The direct response to that has focused on the adoption of information technology (IT) defenses such as passive network monitoring and offline traffic analysis.

Metaphorically, conventional defenses may be viewed as building walls, sometimes referred to as perimeter defenses, such as firewalls, cryptography, message authentication, passwords, etc. In recent years, great strides in defensive measures relying on active defenses and deception-based defenses have been made. These defenses have a common goal, that is, to protect the information from being accessed and tampered with adversaries. Such information includes, but is not limited to, the engineering data flowing into the control network, such as sensor readings, component status indicators, actuator commands, administrative data, surveillance data, etc. Despite effectiveness of those defenses in stopping attacks, they could be compromised if their design details are leaked to attackers. Given the frequency and sophistication of recent attacks, e.g., the 2010 Stuxnet against Iran [1], the 2015 Electric Grid attack against Ukraine [2], etc., state-sponsored attackers have indeed proven that they can acquire proprietary design data relying on a number of techniques, such as espionage, social engineering, insiders' assistance, etc. [3] One of the implications of that is that if they gain access to a control system, and its raw information, the industrial system regulated by those control systems becomes vulnerable and, worse yet, the system could remain defenseless and sustain physical damage.

To address this rising challenge, it has become critically essential to build another layer of defense when IT defenses are compromised. This new layer is referred to as the operational technology (OT) defense [4]. The OT defenses focus on the physical process as described by the network data

comprised of sensors readings, process variables, and actuating commands. The OT defenses ask the question: are the engineering network data consistent with expected behavior? In a sophisticated FDI (false data injection) attack, the attacker relies first on delivering an IT payload designed to penetrate through the IT defenses. This represents the conventional first step for any hacking attempt, i.e., gaining access to the system. Following that, the attacker must deliver another payload, referred to as the engineering payload. This payload is designed to cause the system to move along an undesirable trajectory.

This can be achieved in multiple manners. For example, the engineering payload could falsify the sensors data, causing the control algorithms to send signals to the actuators that cause undesirable performance. Another approach is to change the control algorithm logic to achieve similar goals. In all scenarios, the payload must be aware of the normal engineering checks that exist in the network. These checks are developed by the engineering team to ensure that system is reliably responding to normal process variations. Thus, unlike IT defenses which rely on the use of generic methods to protect access to the information, OT defenses must be cognizant of the engineering design and safety procedures in place. To achieve that, OT defenses must rely on an online monitoring approach to continuously check the engineering data, i.e., sensors readings, process variables, and actuators commands, and be able to determine whether the data are real, i.e., have originated from the system, or falsified, i.e., have been potentially tampered with.

This distinction between IT and OT defenses underpins the key challenge for designing OT defenses. For IT defenses, the goal is to block access regardless of the engineering values of the process variables, implying that one simply needs to adopt a fortress defense mentality. It does not matter what one protects, only how to build an incredible barrier that is difficult to bypass. For OT defenses, however, the goal is to determine whether the variables of the system process variables (an inevitable occurrence in any real industrial process) are naturally occurring or maliciously introduced. The implication is that the defense must depend on a pre-determined specification of what is normal and what is abnormal. Further, given the proliferation of critical systems worldwide, representing the key components of any country's critical infrastructure, the OT defenses must assume that the attacker will likely recruit subject matter experts capable of differentiating between normal and abnormal behavior.

Many approaches have been proposed to design such metrics, often referred to as signatures. The signatures serve as fingerprints for the system, including its physics, and history of operation, where no two systems could be identically the same, even if their initial design is the same. These signatures are designed to ensure consistency and coherency of the process variables used to describe/monitor the physical process.

A key challenge of signature-based methods is the ability to distinguish between normal and malicious behavior under various assumptions of the attacker's familiarity with the system. For example, when the attacker has little or no familiarity with the system, outlier/anomaly detection techniques present the most straightforward approach to detecting FDI attacks [5][6]. In this scenario, each process variable has a prescribed range for variation, e.g., steam generator level, with deviations thereof -- as measured by one or two standard deviations -- indicating an abnormal behavior. This approach has the advantage of being simple to implement, however it does not provide information on whether the abnormal behavior is due to a malfunction or due to an FDI attack.

Next, if the attacker has a basic understanding of the system behavior, outlier/anomaly techniques may not be effective because the attackers might know the preset values that trigger the outlier detection algorithm. In this scenario, another class of methods may be more effective, the so-called data-driven techniques, which rely on building predictive models for the system behavior [7]. Data-driven modeling implies that the physics models are not incorporated to guide the training of the models. Instead, auto-correlation-type regression techniques [8], and their more sophisticated neural-network implementations are employed to predict the present behavior as a function of past behavior [9] . When the predictions made by these models become inconsistent with observed behavior, an alarm is issued. Just like outlier/anomaly detection techniques, data-driven techniques are simple to implement. Also, the data-driven approaches need vast amounts of data, especially for complicated industrial systems, to ensure an accurate emulation of system behavior. Also, they can be customized with reasonable accuracy to recognize different equipment failure modes [10]. This simplicity however means that the learning process can be duplicated by an attacker during an initial lie-in-wait period. This follows because the mathematical machinery for data-driven techniques is well-understood and does not rely on any obscurity measures. Once learned, the attacker can proceed to make changes to the system state that respects the consistency between

present and past behavior. One key disadvantage of pure data-driven learning is that it does not incorporate the physics in the learning process, which implies that if the raw sensors data are routinely falsified, one cannot rely on such methods to detect sophisticated FDI attacks.

In the next level, the attack would be intruded by the attackers who has a general understanding of system behavior but may not be able to exactly replicate it, because they do not have access to key proprietary data and historical operational details. This raises the first research topic that whether these technically-able attackers can learn the system accurately so as to launch attacks within the control limits. Correspondingly, the OT defense is expected to rely on the formal physics description for the system in order to decide what normal behavior looks like. This OT defense is denoted as model-based, since it relies on a physics model to establish a basis for normal behavior. This approach derives its strength from the operational uniqueness and complex interactions between system components.

The next attack scenario, expected to be launched by state-sponsored organizations, the attacker will likely have access to high fidelity simulators for system behavior. For these attacks, referred to as knowledge-based/stealthy attacks, the requirement for an OT model-based defense becomes its capability of detecting signs of FDI attacks when the attackers can predict system behavior to a reasonable accuracy. This represents the focus of this study. This research proposes the use of a model-based approach hybrid with data mining techniques to identify signatures, which can be done by analyzing model simulation results for a wide range of conditions in search of signatures that cannot be identified by the attacker. In doing so, it is assumed that the attacker has access to physics models and data-driven techniques and hence can perform the same job. Specifically, this work shows that the defender can develop signatures, based on the higher-order differences between his/her model and that of the attackers, that are capable of distinguishing between normal behavior and FDI attacks. These higher order effects are typically discarded by most data-driven techniques, and are attributed to sources of uncertainties that cannot be explained by the models. Coupling these higher order effects with dominant behavior can be shown to establish signatures that are difficult to duplicate by the attacker. Clearly, if the attacker has the same model employed by the OT defense, this defense can also be bypassed. This extreme scenario is not considered in this work, and is discussed under the context of active OT defense [11]. The current research focuses on a passive OT defense, where the passivity implies that the defense does not introduce

any changes to the system. It only monitors the measured process variables and compares them to predicted values in search of signatures, as described earlier.

1.2 Problem Statements and Research Questions

As stated in overview, when constructing the signature of physical process, it is natural to rely on the physics governing the behavior of the system, and its mathematical description as embodied in a computer model. The approach is referred to as model-based defense [12], physics-based defense, or with one rendition coining the term digital twin [13]. The basic assumption here is that IT defenses have already been bypassed, and so one has to design another line of defense that can detect manipulation of the raw engineering data used to control the system. Model-based defenses are based on the premise that the defender has access to the most faithful description of the real physical process. This follows because for most critical systems, the models are carefully calibrated to operating data resulting in the best possible prediction capability of the real process behavior, with the measurement noise being the only source of discrepancy between measurements and predictions. Under these conditions, one could rely on a systematic monitoring approach, potentially aided by machine learning and artificial intelligence techniques [14], to look for any other sources of discrepancies between measurement and model-based predictions, representing a basis for detecting unauthorized manipulation of the data. Accepting this premise implies that the attacker cannot predict system behavior to the same level of accuracy as that of the defender. While this assumption could be true for some systems, e.g., highly-classified weapon systems, it certainly can be challenged for critical systems, such as nuclear reactors, chemical reactors, and oil and gas plants.

Focusing the discussion on nuclear reactors, with parallels easily made for other critical systems, one could argue that attackers can acquire models of the same accuracy level as those employed by the defender, either directly or after an initial lie-in-wait period to self-learn reactor behavior. If this is possible, the attacker will be able to modify the raw data in a manner that respects the physics and hence evades detection. For nuclear reactors, such a scenario is not far-fetched, quite the opposite, as evident from the numerous Ph. D. dissertations conducted over the years to accurately simulate reactor behavior over a wide range of conditions, including normal state operation, to anticipated operational transients, design-basis accidents, and all the way to beyond-

design basis accidents, e.g. core meltdown, response to aircraft crash, earthquakes etc. Further, the supply chains for nuclear reactors are extremely diversified including domestic and foreign manufacturers with many individuals involved. Also, there exists a large number of companies which sell reactor simulators that have been customized for existing nuclear reactors, used for training operators as well as for plant control. These simulators are based on faithful replicas of the reactor at all levels, including the I/O level, the PLCs (Programmable Logic Control units) level, the HMI (Human Machine Interface) level, etc.

Given the detailed knowledge widely available about the various nuclear reactors, the effectiveness of model-based defenses should be investigated to determine their resiliency under these extreme adversarial conditions. This work focuses here on the following questions:

1. Can attackers learn the system behavior by relying only on recent advanced machine learning techniques, i.e., without employing physics models?
2. Can attackers learn the system behavior starting with approximate physics models?
3. If attackers with approximate physics models can learn the system behavior accurately and launch an FDI attack without triggering alarm, i.e. within control limit of current anomaly/outlier detection, can model-based defense identify the FDI attack?

All questions will be answered in a virtual sense, where real engineering data, assumed to be accessed by the attackers, are generated using a simulator, considered to be inaccessible to the attacker. We then compare the model learned based on such data and compare it with the actual reactor model. For the first two questions, a preliminary comparison study will provide a directional conclusion towards both of model-based and data-driven approaches. For the third question, a series of scenarios will serve as case study to illustrate the newly proposed algorithm.

2 LITERATURE REVIEW

2.1 Current strategies adopted for FDI Detection

The concept of false data injection (FDI) attack was originally introduced in the smart grid domain. Specifically, it refers to the case when an attacker corrupts sensor readings in such a stealthy way that undetected errors are propagated into the controllers and thereafter lead to miscalculation of the state variables [15]. Due to the prosperity of the digitalization and associated complex control systems, cyber attackers are interested in exploiting this type of attacks in other industries and domains, such as nuclear industry. Unlike traditional cyber-attacks that target data availability or confidentiality, such as denial-of-service, jamming, etc., false data injection (FDI), targeting data integrity, is one of the most frequent, realizable and lethal attacks, since the attack vectors could be simple as a constant value or trapped as a period of plausible signals. Specifically, the FDI attacks is able to circumvent anomaly detection such that the injection measurements will be undetected. In today's increasingly perilous cyber world of complex adaptive systems, FDI attack has become one of the top-priority issues to counter. It is a necessity for strengthening awareness and a more sophisticated mechanism to address this attack in the cyberspace. To address this issue, scholars in the nuclear industry have proposed various strategies to detect and respond to the FDI attacks.

The detection process is accomplished by monitoring system, which can be briefly categorized into two types: passive monitoring and active monitoring. In the context of FDI detection, the passive monitoring refers to the techniques observing system behavior in search of patterns of normal behavior with deviations thereof representing abnormal behavior, without making any changes to the system. Distinctively, the active monitoring, known as synthetic monitoring, refers to the other type of techniques involving testing packets injection into the system and then measuring its performance to authenticate the system status [16] [17]. Without doubt, adoption of both active and passive monitoring would optimize the performance of the control system, but one cannot make the most of active monitoring without a good understanding of the physics model, which composes an essential content of passive monitoring. As the cybersecurity of nuclear control system is still in its adolescence, studies in either active monitoring or passive monitoring are not amplified. Therefore this Ph.D. work mainly focuses on passive monitoring with physics insights

involved. Passive monitoring OT defenses adopt two general approaches: data-driven or model-based approaches. Data-driven approaches rely on the use of data mining techniques to establish patterns in the engineering data based on past behavior. An adversary trying to change the engineering data, e.g., process variables and sensors data, unaware of these patterns, will introduce changes that can be detected when deviating from the patterns. Model-based approaches require building a faithful mathematical model that describes the behavior of the system. When the measurements deviate from the predictions, alarms may be issued. Next, we overview some of the methods employing both data-driven and model-based techniques from the literature.

Eggers employs principal component analysis (PCA), independent component analysis (ICA), and their variants to detect FDI attack intrusion under different scenarios. This work indicates that static ICA and PCA may not be sufficient, but a moving window version of PCA and ICA can quantitively identify the onset of the attacks [18]. Her work has primarily focused on FDI attacks that introduce noticeable changes to the engineering data, e.g., a sudden drop, patterned sensor drifts, or flat-lining of process variables. Zhang et al. employs auto-associative kernel regression (AAKR) to determine whether a sensor data is authentic by using the idea of sensor virtualization, wherein a group of sensors are employed to predict the reading of a given suspect sensor. The residual between the measured and predicted values serves as a measure of authenticity, i.e., if the residuals go beyond the predefined threshold, an SVR-based inference model is employed to calculate the countermeasure commands sent to actuators [19].

W. Wang et al apply a nonparametric cumulative sum (NP-CUSUM) approach to the Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED) for online monitoring of cyber-attacks to on a multiloop-controller. [20] In that work, the detection is fulfilled by a constructed score function, which will not exceed the prespecified threshold under normal operation. This approach is proved to be effective with the practice of the four attack scenarios. Instead building a score function, H.L. Gawand et al employ least square approximation followed with convex hull approach to determine the authenticity of the measurement [21]. It is effective for FDI attacks that manifest as random bias.

Y. Zhao et al. employ a competitive Markov decision process to model the interactions between the defender and the attacker, in which credible state transition probability is provided by

probability analysis (PRA) [22]. In this study, each state is quantified as a state value, which is taken as reward for defenders and attackers, and the best choice is calculated based on the rewards value. This study provides a guide for defenders' optimal response to cyber-attacks under various situations. Similarly, based on the state transition probability from PRA, P. K.Vaddi et al. employ the dynamic Bayesian Network for inferring the hidden state of the system from the observed variables by probabilistic theory [23]. Both works are built based on the state transition probability derived from probability risk analysis (PRA), indicating that the success and credibility of the whole model highly depend on the reliability of these prior knowledge.

Liu, et al. show that attackers are capable of constructing attack vectors within a constraints in order to change state estimations without triggering the alarm, referred to as stealthy FDI[15]. R. Smith employs linear and nonlinear physics models to illustrate detectability of stealthy FDI attacks with respect to operating point changes, and confirms that when the attackers have more sophisticated resources, the probability of keeping attacks undetected will correspondingly increase [24]. Sandberg, et al. employ convex optimization tools to evaluate attacks, by taking deviations from the true model and attack goals to quantify the least efforts needed to achieve this type of attack, i.e. the minimum number of needed compromised sensors in a certain system [25]. The same research goal for attack evaluation is fulfilled in [26], the results of which indicate that information related to operating conditions and saturation limits is necessary for successful stealthy FDI attacks on nonlinear model. Beside, a generalize approach to construct FDI attacks with specific constraints on state estimation is proposed in [26].

Other researchers have focused on measuring the consequences of different compromised components. Kosut et al. [27] firstly introduced the concept of 'strong attack regime ' and 'weak attack regime', the first of which refers to the attack scenario with a large number of compromised meters to keep the attack unobservable. This work employs graphic theoretic method to determine the smallest set of compromised meters that the attackers need to manipulate the system to hide the attack. Similarly, the 'weak attack regime' refers to a smaller set of meters than that in 'strong attack regime'. In this study, the trade-off between raising the state estimation error and reducing the detection probability is investigated. O. Vukovicet al. study several common attack vectors; investigate how a single compromised control center can affect state estimation by tracking the evolution of the number of outliers state estimations [28]. Meanwhile, some studies focus on the

signature development based on data-driven/physics-based defenses. Hadžiosmanovi´c et al. perform a data characterization phase approach based on the data behavior: continuously change, state reflection, or constant. These different groups of data are used to fit an autoregressive model, which aims to estimate the behaviors of a correlated system state [8]. Since the intrusion detection algorithms for SCADA systems look at anomalies in state-based estimation, these approaches are usually poorly adopted to process data. Quinn and Sugiyama used a least-squares approach to detect anomaly in static and sequential data, but this research was not specifically looking at cyber security anomalies. [29] Krotofil and Cardenas studied the resilience of an industrial control system to a cyber-attack using the Tennessee Eastman challenge process in order to develop a systematic approach to cyber security assessment of ICSs and analyzing the effects of hypothesized cyber-attacks. [30] In addition, game theory techniques were proposed as intrusion detection methods as a reactive responses in CPSs. [12] Given the lack of practical experiments, data-driven algorithms developed for online equipment condition monitoring may prove to be the most useful algorithms for detecting false data injection attacks in NPP process data.

Other researchers have focused on developing requirements to establish effective countermeasures for fighting FDI attacks. As few examples, Dan, et al. propose two data-driven algorithms to study the costs of specific attack and the cost for implementing defenses [31]. Giani, et al. introduce and characterize irreducible cyberattacks to identify the minimum number of needed known-secure sensors to disable FDI attacks [32]. Kim, et al. suggest a subset selection algorithm to identify the key measurements to be protected when the defenders have limited resources, based on constructing attacking vectors developed for linearized measurement models [33].

For employing the model-derived correlations, Li et al. employ dynamic PCA to characterize the correlation between multiple variables and consequently use Chi-square detector to distinguish adversarial cyber-attacks from ordinary random failures [34]. Urbina et al. suggest a physics-based attack detection algorithm, which aims to set up a proper error threshold for the sensor within a certain time period [12]. Their algorithm is proved to be capable of adaptive adversary attacks. Related works can be found in [35][36][37]. For active monitoring, A. Sundaram et al. propose a data analytical approach, which introduces noise into the network, and hence detect unauthorized manipulation via assessment with regard to the impact on the system. [38] Y. Zhao and C. Smidts

employ a two-step Chi-square hypothesis testing method with physical watermarking for detecting and distinguishing replay attacks from other anomalies [39].

Table. 1. Summary of taxonomy of FDI detection techniques in nuclear industry

	Application	Approach	Technique category
Passive Monitoring	Steam Generator of a 2-loop PWR simulator[19]	AAKR	Data-driven
	ALFRED [20]	NP-CUSUM	Data-driven
	Digital feedwater control system [22]	Game theory (Markov decision process)	Model-based
	SBLOCA of PWR [18]	PCA and ICA	Data-driven
	Digital Feed water control system [23]	Dynamic Bayesian Networks	Model-based
	Feed water control system [34]	Chi-square detector and dynamic PCA	Model-based
Active Monitoring	Steam turbine system and Gas turbine system [38]	Colored noise detection	Data-driven
	Steam Generator [39]	Physical watermarking	Data-driven

2.1.1 Contributions of previous studies

There is a growing literature on the security of CPS, including the wide application scenarios like power plants, electronic devices and so on. In the first place, previous works proposed various detection methods and validated these methods with testbeds, comparison with credible simulation codes etc. These methods build a library for scholars to find effective detection approaches and spark appearance of novel approaches. Based on these works, scholars integrated and subtract essence of the current methods with unified taxonomy based on different topics, such as the applied venue, attack location, validation metrics, detection algorithms, etc., which allows identification of limits and unexplored challenges, and eventually develop a framework to accommodate the various methods. [12]

Specifically, for FDI detection techniques in nuclear industry, most studies, including both passive and active monitoring, take advantage of the easy availability of historical monitoring data via implementation of data-driven approaches. [18][20][40]

2.1.2 Limitations of previous studies

In this section, the difficulty levels on detecting FDI attacks depend on the attackers' knowledge about the system. Specifically, as the nuclear simulation has been developing for several decades, almost all kinds of simulators for different types of nuclear reactors/systems and the corresponding validation studies can be found via open access, such as Ph.D. thesis, published reports and research papers. Especially for commercial nuclear power plants whose safety and maturity have been proved/validated over decades, the insider/outsider attackers have sufficient resources to be familiar with the physics modeling. Most of current work to detect the FDI attacks are constructed via the data-driven techniques, which aim to learn the system behavior based on historical data. With the rapid development of machine learning techniques, it is feasible for attackers to implement different learning algorithms like various kinds of neural networks to learn the system details starting with an approximate physics model. Few current works assume attacker's familiarity towards the nuclear systems, which may not be efficacious when dealing with stealthy FDI attacks.

Model-based detection has appeared in nuclear cybersecurity in recent years. Some works are based on prior knowledge, such as the state transition probability from PRA employed for game theoretic detection approach or Bayesian neural network. [22][23] However, for this type of approaches, the success and credibility of the whole model highly depend on the reliability of the prior knowledge. Either an overly optimistic or pessimistic expectation of the quality of these prior beliefs will make the entire network misrepresent the true physics and so nullify the results. Besides, the PRA model can be found via open access, in other words, the whole structure of the probabilistic analysis can be learned by attackers as well. With this knowledge of the probabilistic network, attackers can reproduce the model via various AI techniques. Specifically, game theoretic approaches assume (1) all player act rationally and intelligently; (2) the rules of play are known to all the players. If we say the first assumption is more or less impractical, the second assumption would be dangerous to defenders, since for well-resourced attackers, both physics

model and general defense strategies are known, but defenders have little knowledge about novel/creative threats from attackers. Meanwhile some scholars include physics insights into the model construction, by identifying the relationships between various process variables, and the FDI detection is based on certain relationships. [34] This type of approaches is sufficiently effective for the attacks from the attackers with limit knowledge of the system, who do not know the certain physics relationships. As stated above, attackers can learn the system accurately with leverage from an approximate physics model and current learning techniques. Also, the relationships usually do not show a clearly highly correlated pattern. For example, the pressure and temperature of a component in the nuclear system are highly correlated as PV = nRT, but their online monitoring data may not show the same clear correlation, since in nuclear system, (1) phase change will highly effect both quantities; (2) the error from measurements are not neglectable; (3) the physics equation describes the steady state, so the dynamic variations may contain time delays; (4) normal operations contain other uncertainties. Due to these reasons, all the possible values for the correlated variables in normal operating conditions would appear as a quite large region instead of a clear line or curve, which provides potential hides for FDI attacks. Thus, a more delicate detection method is needed to identify subtle monitoring measurement changes.

2.2 Strategy of this work

These aforementioned methods, developed in different disciplines, share one thing in common: they rely on capturing the dominant behavior from operational data or correlation between process variables, mathematically referred to as active degrees of freedom (DOFs) or lower order components (LOCs), to make predictions. For example, PCA relies on singular value decomposition (SVD) to compute the principal components, representing the few right or left singular vectors, referred to hereinafter as the LOCs.

For illustration, Figure 2.1 shows the components of a typical sensor variations as projected onto the components identified by PCA. The *x*-axis represents the index of the components and *y*-axis shows the significance of the components. The components in the blue box represents the LOCs, expected to be known by the attackers, which can be captured by data-driven techniques or an approximate physics model [15]. This implies that, once the LOCs are learned during initial lie-in-wait period, the attackers would be able to falsify the measurements respecting the dominant

patterns of system behavior. With that, they would be able to bypass detection by techniques relying solely on capturing the dominant behavior [41]. The LOCs however remain an effective signature for patterns that significantly deviate from normal behavior such as: (1) a constant bias, (2) a measurement drift with a function of time, (3) wider noise, (4) dynamic process variable freezing as a constant, etc. If the attackers however attempt to falsify the data using the LOCs, the majority of FDI techniques described earlier would be potentially bypassed.

To detect these stealthy attack scenarios, one need to rely on capturing more features that capture the less-dominant patterns of system behavior, denoted hereinafter as higher order components (HOCs), as shown in the yellow box in Figure 2.1. Unlike the LOCs that can be seamlessly captured by approximate models, the HOCs are much more sensitive to the system characteristics, e.g., past behavior, modeling assumptions, etc. If the attacker has the same exact model as the defender, then ultimately the proposed approach may also be bypassed. As mentioned earlier, this extreme scenario should be handled using active monitoring which is outside the scope of this work. Instead, it is assumed that the attackers are equipped with a sufficiently accurate model, however it does not faithfully duplicate the defender's model, often carefully calibrated to operational data. As shown later, the HOCs allow the defender to take advantage of the subtle variations between the attackers and defenders' models, allowing defenders to detect FDI attacks that respect the patterns established by the LOCs. The idea is to derive features that are based on both the LOCs and HOCs as a basis for classifying normal from FDI behavior masquerading as normal behavior.

This work firstly proposes the use of randomized window decomposition (RWD) to identify the LOCs and HOCs-based features [42], which has demonstrated the potential of this OT defense for an idealized scenario, where the data were assumed to be available offline. Then this work focuses on how it can be used to detect subtle data falsifications in real time with interference from normal process noise. Particularly, a stealthy attack called triangle attack [43] which employs a series of line segment(s) to respect the system dynamic behavior without prior knowledge of the system dynamical model. The RWD algorithm is adapted for real-time and is equipped with a denoising algorithm to ensure noise does not interfere with the HOCs.

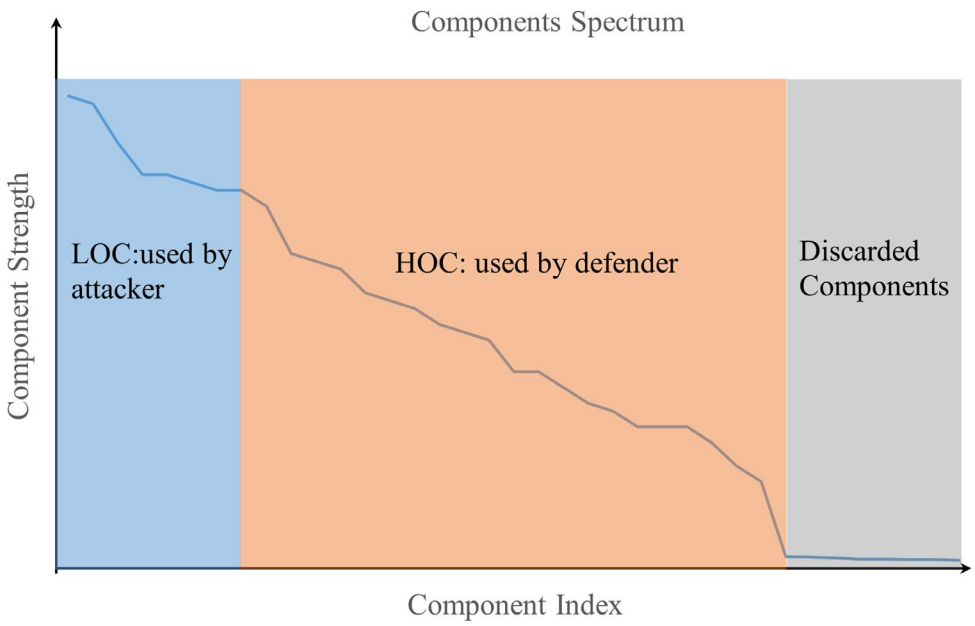

Figure 2.1 Dominance Ordering of Extracted Process Patterns

2.3 Contribution and Limitation of this work

This work proposes an online monitoring technique that can identify subtle monitoring measurement changes. Unlike traditional methods that focus on the most dominant behavior of the system, this work exploits the information of the less dominant behavior serving as fingerprints of the physics model, since previous studies have proved that well-resourced attackers are able to have access to the most dominant behaviors. [44] The experimental work in this dissertation mainly consists of two major parts. The first part is denoted as a preliminary study and the other part is named as an exploratory study. In the preliminary study, two case studies are employed and provide proof for:

1. Solely with historical operation/monitoring data, the attacker can learn the dominant behavior of the physics system.

2. With historical operation/monitoring data and an approximate physics model that can be reached via open access, the attacker can learn the missing information of the approximate physics model, i.e. model parameters, and consequently learn the system accurately.

Besides, the current FDI detection approaches mainly focus on addressing sudden changes. To cope with the subtle FDI attacks/system variations, this work proposes an algorithm with extracted new feature, supported by a new denoising method, which is demonstrated in the exploratory study, representing the focus of this study. In the exploratory study, the detection algorithm firstly works as an offline technique to detect triangle FDI attack, component degradation and accident, and then it develops into an online monitoring toolkit. The offline technique usually takes a long snapshot to conduct a detection algorithm, since the goal of offline monitoring is to detect the attack and the longer window contains more information and is more robust to noise. However, online monitoring requires a smaller window size to enable fast turnaround times for the executions of detection algorithm and the possible following countermeasures against attacks. Consequently, a denoising approach is needed to cooperate with the detection of subtle triangle attacks. Commonly used denoising techniques would smooth out the HOCs together with noise. Thus, a novel denoising method is proposed under the same mathematical theoretical frame as the detection algorithm to support the subtle FDI detection. In the exploratory study, the mathematical developments contain three parts: (1) offline monitoring technique; (2) denoising approach; (3) online monitoring toolkit.

The aforementioned assumptions that the attackers have no access to all proprietary design details or full library of historical data, reveal the one limitation of this work that this approach can only identify the system behaviors that deviate from the patterns/structures within the genuine pattern library. For the attacks employing genuine data as the injected signals, e.g. replay attack, the methods in this work will not be effective. Besides, since the detection algorithm is based on the variation of the relationships between LOCs and HOCs, the attack vector in the null space of the identified components will not be identified, which represents another limitation of this work.

3 BACKGROUND

With the definition of problems to study, this section provides a description of research object, industrial control system, and previous works focusing on detection of FDI attacks.

3.1 Background in Industrial Control System (ICS)/Cyber Physical System (CPS)

At a high level, any industrial control system may be considered as containing four major parts as illustrated in Figure 3.1. Actual industrial control systems will typically include multiple instantiations of those parts, which enables the system capable of cognition, communication, computation and control, denominated as 4C [45].

Physical process: it represents the core of the system, i.e., the release of energy following fission of nuclear fuel, and resulting in the establishment of neutron flux, heating of the fuel, and transfer of heat to other parts of the system. This physical process generates a physical response p_n representing a change in the system state, e.g., fuel temperature increase/decrease, coolant temperature, neutron flux, etc.

- Sensor: it is placed in the system to sense a change in its state, and produce a signal, typically analog, which is converted into digital form, denoted by y_n. No distinction is made here between analog and digital signals, since it is currently outside the scope of this article. This signal is sent to the next component, the controller.

- Controller: it receives the sensor signals, performs some initial data processing/testing to remove noise, detects outliers, ensures physical consistency, etc., and then employs the control logic to calculate a command to change the reactor state using actuators. Such change is executed to achieve a certain goal, e.g., maintain current power, ramp up power over a given time period, etc.

- Actuator: it converts the control commands into physical changes, e.g., movement of control rods, partially opening or closing a valve, etc.

This work does not discuss how the attackers gain access to the system, and how they will inject falsifications to the sensor readings or actuators. For our purposes, we assume that they can inject the attacks as direct perturbations to the sensor readings and/or actuator commands, referred to as false data injection (FDI). In principle, they can also attack the controller's logic as well, but these details will be left to future work. Mathematically, this may be described as follows:

$p_n = P(u_{n+1})$;

$y_n = S(p_n)$, if genuine, or $y_n^{FDI} = S(p_n) + \Delta_y^{FDI}$ if under an FDI attack targeted at the sensor readings.

$u_{n+1} = C(y_n)$, if genuine, or $u_{n+1} = C(y_n^{FDI})$ or $u_{n+1}^{FDI} = C(y_n) + \Delta_u^{FDI}$ if under attack.

The latter attack scenario can be introduced either by changing the sensor readings causing the control system to take the wrong action, or by directly changing the control actions. One could envision a hybrid approach where both sensor readings and control actions are falsified.

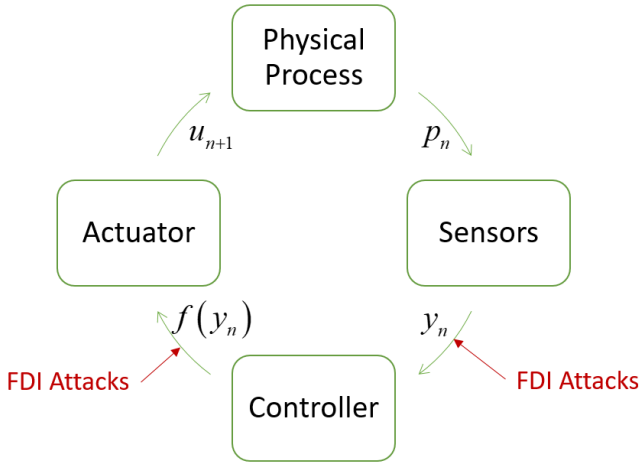

Figure 3.1 Generic Industrial Control System

3.2 Vulnerability of CPS

3.2.1 Cyber Attacks in Nuclear Fields/Industry (Timeline)

Recent world is filled with examples indicating that critical infrastructure, like nuclear facilities are prone to both ransomware and targeted cyberattacks. Back in 1992, a computer programmer at the Ignalina Power Reactor Station in Lithuania inserted a virus into one of the stations computers attempting to sabotage a reactor at the plant [46]. Later on in 2003, the MS SQL Server 2000 worm has infiltrated the Davis-Besse nuclear power station. The infection led to data overload in the network, resulting in the failure of the computers to communicate with each other. [47] A former employee of a Texas power utility programmed the models which regulated the management of EFH power generation facilities to cripple the company's energy forecast system in May 2009 resulting in financial losses. [48] As is now well known, between 2009 and 2010 the Stuxnet virus targeted the Natanz uranium enrichment facility in Iran [1]; The virus triggered weakened centrifuges and disrupted enrichment operations as well. In fact, this situation is noteworthy because the facility was well defended and disconnected from the Internet. Revelations of malware discovered in nuclear installations and critical infrastructure have risen in volume after news of Stuxnet broke in 2010. In 2014 alone, The German Steel Mill was infiltrated by malware which employed spear phishing email w to obtain access to the corporate network and then

transferred into the plant network; [49] malware was introduced into the control room at the Monju nuclear power plant in Japan; [50] cyber attacks against the Korea Hydro and Nuclear Power in South Korea led to leak of a blueprint and details of various support systems. [51] [52] A Japanese facility that processes plutonium and other nuclear materials disclosed that it had detected malware in its systems in 2015. [53] In 2016, Gundremmingen nuclear power plant in German reported to be compromised with ransomware designed to encrypt files from hacked computers [54] A former employee of the DOE and the U.S. NRC attempted to sell the information which was proclaimed useful to inject a virus on NRC computers that could allow access to agency information from foreign countries or could be used to otherwise shut down the servers of the NRC. [55] In 2017, a sophisticated, troubling cyberattack using the Petya ransomware was launched against the Ukrainian power grid that temporarily disrupted the electricity supply to consumers for a period from one to six hours. [56] Based on the function of Petya, a destructive malware named as NotPetya was enhanced to spread broadly and was believed to specifically target Ukraine. [57] In 2019, an Indian nuclear reactor was infiltrated by North Korean attackers with leakage of huge amount of operational data. [58] The detailed description and collection of the previous attacks can be found in [3][59][60][61][19]. Based on previous collection, this work adds most recent attack incidents as shown in the timeline in Figure 3.2.

From the cyber-events the world witnessed so far, the technological skills of threat actors have vastly enhanced and their ability to inflict physical harm is surprising. Stuxnet, for example, indicated that cyber-media would have a huge effect on the real universe. Stuxnet was an incredibly sophisticated cyber-attack carried out using specialized malware that targeted a particular ICS. A crucial lesson learnt from Stuxnet is that whatever device it needs will certainly be attacked by a well-financed advanced threat agent, which can create alarm for critical infrastructure. For critical infrastructure, the most important lesson to be learnt is improving the ability to detect and recover from a cyber attack, since it is not possible to defend all networks from any intruder. We learnt from attacks that basic tactics are enough to hack through sensitive networks, implemented by a professional and persistent adversary.

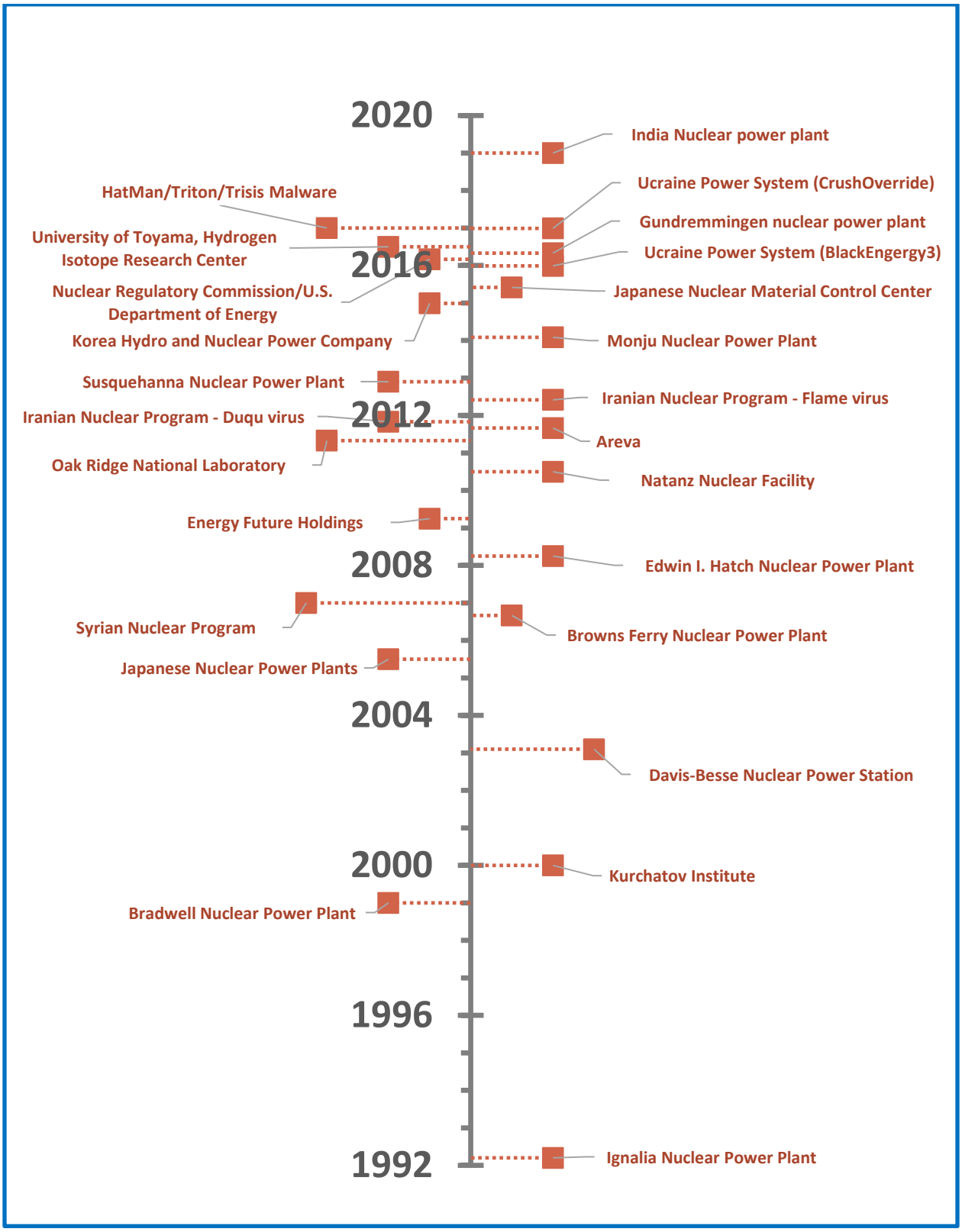

Figure 3.2 Cyber Attacks against Nuclear Industry

3.2.2 Risk Analysis of CPS in nuclear industry

Nuclear facilities are vulnerable to a variety of cyberattacks by a variety of malicious attacks. In addition to the recent CPS attack accidents in the nuclear industry, several scholars have undertaken CPS studies, derivation, and risk identification and control.

T. MacLean et al. examine the existence of potential attack vectors in an NPP; specifically, they build a testbed to simulate a boron monitoring system and an FDI attack on the programmable logic controller (PLC). Their results indicate the vulnerability of the control system to the attack simulated by easily available software and hardware. [62]

Researchers also put endeavor to identify potential vulnerability in a nuclear system. Researchers perform attack vector analyses based on the RG 5.71 which provides a complete set of requirements for the cyber security of NPP I&C systems. They analyze the architecture of the CPS of NPP in the first place and provide a list of vulnerability and potential penetration. And they consequently propose requirements par each possible vulnerability of the industrial control system. [63] [64][65] Varuttamaseni et al. focus on building a cyber attack model for nuclear power plants. The propagation of the attack is modeled by considering certain attributes of the digital components in the system. These attributes help the identification of the potential vulnerability of a component to different classes of attack and the capability gained by the attackers once they are in control of the equipment. [66]

3.3 Official Guide for development of cyber security for nuclear system (Timeline/frame structure)

For the nuclear energy industry, safety is paramount. It is protected by multiple back-up safety systems, robust physical defenses, and plant security forces which undergo rigorous training and preparation for emerging threats, including natural disasters, cyberattacks on critical operational systems etc.

The nuclear sector has suffered from various cyber security issues for a long time. Back in 1997, the industry embarked on the investigation of potential issues associated with the growing usage of digitalization at power reactors. [67] Specifically, the Nuclear Regulatory Commission (NRC)

is the governmental body in the US that formulates policies and develops federal regulations for NPPs. In 1997, NRC had issued a series of complimentary documents to resolve the issue of the applied digital software in NPP, DG–1206~1210. [68] During the years 2003 and 2004, the nuclear industry embarked on the regulations to support the standard deployment of cyber security systems at nuclear reactors. Four nuclear power reactors in the United States completed cyber security assessment pilots in July 2003.

These pilots were devised to inform development of NUREG/CR-6847, “Cyber Security Self-Assessment Method for U.S. Nuclear Power Plants”. [69] The project team was made up of representatives from the Pacific Northwest National Laboratory (PNNL) and the Nuclear Regulatory Commission (NRC), released in November 2004. The guidance outlines a risk-informed approach that takes into account the effects on plant functions, as well as how to develop a plant-wide cyber security defensive framework that enables various defense layers with escalating levels of security protection.

The nuclear industry set up the Nuclear Strategic Issues Advisory Committee (NSIAC) in December 2005, compromised of the Chief Nuclear Officers of each nuclear power plant site or fleet, which is capable of establishing initiatives that are binding efforts for all nuclear power plants. The NRC requires power plants to develop, implement, and assess physical and cyber security plans in order to protect against a Design Basis Threat (DBT). In 2007, the NRC revised the DBT specification to include a cyberattack as an attribute of the adversary in response to the growing threat of cyber-related threats. The NRC published new security requirements in March 2009, which included thorough programmatic cyber security measures defined primarily in Title 10 of the Code of Federal Regulations (CFR), Section 10 CFR 73.54, titled as “Protection of Digital Computer and Communication Systems and Networks”. [70] The regulation requires nuclear power plants to submit a cyber security plan and implementation schedule for NRC review and approval. In April 2010, NRC approved NEI 08-09 [71], which provides a cybersecurity management for nuclear power reactors intends to facilitate nuclear power industries in complying with 10 CFR 73.54, as well as a catalog of technical, operational, and management cyber security measures adapted from NIST Special Publication (SP) 800-53, "Recommended Security Controls for Federal Information Systems". Serving as implementation milestones, this template includes identification of critical digital assets, alleviation of cybersecurity controls and examination of

cybersecurity practices, etc. With regards to the vulnerability of the digital ICS and the intermittent attacks against nuclear systems, NRC issued 10 CFR 73.77 "Cybersecurity Event Notifications" in Nov. 2015, requiring NPPs to record and report cyber security events. [72]

3.4 FDI Attack Types

A literature review on the construction, detection and assessment of FDI attack against CPS is conducted in Chapter 2. Here the existing attack vectors injected in time series have been categorized into several types according to their trend and the construction of the types.

- Freezing Attack Vector

As stated in 3.1, the genuine signal displayed in controller is expressed as $y_n = S(p_n)$, where y_n represents the genuine signal at the n^{th} time step and $S(p_n)$ represents the corresponding measurements from sensor. Freezing attack vector, expressed in the Eq. (1), is the most popular attack vectors studied in the field of cybersecurity of nuclear industry. [18][40][20] In (1), C is a constant value that could be a previous state value of the process variable or an arbitrary state value. This type of attack vectors has been proved detectable via different detection algorithms, e.g., data-driven techniques, residual based approaches etc.

$$y_n^{FDI} = C \tag{1}$$

- Recurring Pattern Attack Vector

Recurring pattern attack vector can be considered as a sophisticated version of the freezing attack vector, instead of replacing the original signals by a constant but a recurring pattern, which can be expressed in Eq. (2), where the function of the recurrent pattern is denoted as F. Specifically, F could represent a saw pattern, sinusoidal or a defined periodic function.

$$y_n^{FDI} = C + F(p_n) \tag{2}$$

- Bias Attack Vector/ Shifting Attack Vector

Bias attack vector can be expressed in Eq. (3), *C* representing a constant shift for all state values after the falsified data intrusion. Also, for different periods of time, the constant can be different, which results in a step function as the attack vector.

$$y_n^{FDI} = S(p_n) + C \tag{3}$$

- Magnified-Noise Attack Vector/Scaling Attack Vector

As the real-time measurements contains noises, the magnified-noise attack vector usually does not change the mean or median of the measurements, which could circumvent the basic statistical check of the monitoring system. This attack vector is expressed in Eq. (4), where $S^0(p_n)$ represents denoised or model inferred measurement at nth time step, δ_n and δ_n' represent the original noise and the magnified noise respectively. Generally speaking, the mean of δ_n and δ_n' is 0, while the standard deviation of δ_n' is larger than δ_n' .

$$\begin{aligned} y_n^{FDI} &= S^0(p_n) + \delta_n' \\ S(p_n) &= S^0(p_n) + \delta_n \end{aligned} \tag{4}$$

- Sensor drift Attack Vector

Sensor drift attack vector (also denoted as drift attack vector) broadly covers a series of attack vectors with a defined function for attack vector construction, whose mathematical expression is shown in Eq. (5), where *f* represents a drift function introducing rapid variation of the process variable. Due to this property, drift attack can be easily detected since the value of process variable would exceed the control limit in a short time.

$$y_n^{FDI} = S(p_n) + f(p_n) \tag{5}$$

- Triangle Attack Vector

Triangle attack is proposed in [43], which aims to find a series of line segments to adjust the local signal variations. The attack vector is established by sending two rays being sent from an identified vertex till the next vertex is reached. This line-segments fit provides a good estimation of a

dynamic process, which could result in very plausible attack to circumvent monitoring system since the local signal variation pattern in kept in the attack vector. For example, while a process variable, denoted as PV_1, keeps decreasing, another process variable, PV_2, is manipulated by reactor practitioners, where PV_1 and PV_2 are correlated. If the attacker aims to make the operator incognizant of the reduction of PV_1, he/she has to preserve the variation of PV_1 resulting from the manipulation of PV_2, while removing the decreasing trend of PV_1.

3.5 Detection Techniques

A physical process defense offers a new security approach that is fundamentally difference IT defenses. This strength of physical process defense is derived from the uniqueness and sophisticated interactions between three essential elements compromising any cyber physical system: (1) Dynamics of the physical process and all variables associated therewith, such as specific design parameters, monitored process variables like sensors readings, actuator commands, components status indicators; (2) the computations, the modeling and simulation tools used for state inference and control; (3) network communication, including network architecture and the employed protocols. This complexity can be leveraged to design equally complex defense measures capable of identifying unauthorized manipulation of system state even when the system remains digitally penetrated.

Scholars have shown that physical process defenses can employ the well-developed mathematical arsenal of data mining and artificial intelligence (AI) techniques to identify the signatures/features (mathematical functions) that serves as fingerprints for the physical system. The detection techniques adopted in CPS monitoring with the signatures can be grossly categorized into three types. The first type is based on the discrepancy between measurements of monitored process variables and the inferred measurements from a simulation model. For the second type of detection approaches, the signature is constructed from the original measurements/observations in form of vectors or tensors as the preprocessing for machine learning techniques. For example, given a series of datasets/measurements combination as vector, one can employ support vector machine (SVM) to classify or regress the vector to the responses, or employ principal component analysis (PCA) to identify the most dominant directions among these vectors. Given the state values of certain process variables, one is not required to provide physics insight into the construction of the

state variable vectors, therefore these approaches are usually referred to as data-driven. The third type of signature is denoted as model-based, since the model information/physics insights are involved while implementing the detection techniques. One way to involve the physics information is to build the signature from a physics perspective. For example, one can apply fast Fourier transform For example, one can employ PCA to obtain the degrees of freedom (DOFs) and consequently select the DOFs and treat them as regressors to build an inference model. Albeit data-driven techniques adopted, the signatures are not plain combination of raw data, but processed with data-driven techniques to find the model information, i.e. the DOFs from implementation of PCA and the selection of DOFs also requires expertise on the physics. A summary of the detection techniques is shown in Table 1.

Table 1 Summary of taxonomy of related detection techniques in control system

	Definition	Peculiarity	Methods
Basic statistical check (outlier/anomaly check)	Identification of measurements that do belong to a certain population	• Easy to implement. • Can be bypassed if the threshold is known	Error analysis
			Correlation Analysis
Data-driven techniques	Detection process is compelled by experimental data instead of physics	• Physics models are not incorporated. • Generic and can be duplicated by attacker. • Requires vast amount of data. • Requires information quality of data. • Mostly applied on black-box problems	SVM
			PCA/ICA
			Bayesian Networks
			Autoencoders
			Neural Networks
			Auto-correlation regression
			Random Forest
Model-based	The predictive model construction employs prior knowledge or involves physics insight	• Derived from physics or with physics insights. • Customized for different physics model.	Kernel PCA
			FLDA
			Kernel FLDA
			Supervised PCA

3.5.1 Basic statistical check

The first type is a basic outlier/anomaly check, which identifies the measurements which do not correspond to normal operational behavior, usually by limiting the discrepancy between the

measurements, y_i, where i represents the index of measurements and the estimated process variable value at i[th] time step, $\hat{y}_i$, within a bound δ. For a period of time with n time steps, one can get the mean value of the measurements, denoted as $\bar{y}$. The mainstream metrics to calculate the discrepancy include but are not limited to the following methods. [45]

Table 2 Metrics of Basic Statistical Check

Mean-squared error (MSE)	$\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}$
Mean absolute error (MAE)	$\frac{\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert}{n}$
Maximum error	$\max\lvert\hat{y}_i - y_i\rvert$
Root mean squared error (RMSE)	$\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$
Sum of squared regression (SSR)	$\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$
Sum of squared error (SSE)	$\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$
R^2 correlation coefficient	$1 - \frac{SSE}{SSR}$

3.5.2 Data-Driven Techniques

The data-driven models purely rely on data and preclude the knowledge of the CPS dynamics. Attempts for data driven modeling for capturing the dynamics of CPS try to exploit the following concepts [73][74].

- K Nearest Neighbors

The most commonly used data-driven technique is kNN due to its fast implementation and capability to handle unsupervised data. For the k-nearest neighbor (kNN) technique, the fundamental assumption is that the genuine data exists in compact cluster and that deviations arise at a distance from the cluster. The number of nearest neighbors within a given distance is calculated by a density dependent nearest neighbor process. For example, the local outlier factor (LOF) is a mechanism in the dataset that manages different densities [74]. A LOF score is calculated as the

ratio of the instance's average local density of kNN and the test instance's local density. A lower LOF score distinguishes the anomaly because the anomaly would have a local population lower than that of its closest neighbors [74].

- Bayesian networks/Bayesian learning

Bayesian networks [75] employ a set of variables and their conditional dependencies via a directed acyclic graph (DAG), a hierarchical model, whose nodes represent variables in the Bayesian sense: observable quantities, latent variables, unknown parameters or hypotheses. These quantities and connected by a set of prior probabilities. Based on this nature, Bayesian networks can be used to find probabilistic queries of the variables, and the learning of the variables. However, one needs to pay extra attention when choosing priors for a hierarchical model, especially on the variables at higher levels of the hierarchy. In nuclear fields, probability risk analysis (PRA) shares a similar hierarchy structure with the Bayesian network, which provides the prior probability for building Bayesian networks [23].

- Autocorrelation regressions

Autocorrelation refers to the correlation across various observations in the data the same variables. In the form of time series data in which observations occur at multiple points along the time axis, the principle of the autocorrelation is most frequently debated. [76] In reality, the data would be autocorrelated if the observations of a process variable that occurs closer in time would be more similar than the temperature values that occurred farther apart in time. In a regression analysis, autocorrelation regression residuals would be employed as a metric to determine whether the model is incorrectly specified. For example, if one attempts to model a simple linear relationship but the relationship observed is non-linear, then the residual may not be autocorrelated.

- Random forest

The random forest is an ensemble technique that can also be viewed as a kind of predictor of the nearest neighbor. Ensembles are a divide-and-conquer technique used for performance enhancement. A community of decision trees, referred to as "small learners", will combine together to build a "strong learner", which is the core concept behind ensemble approaches [77].

Random forest is employed as a robust and easy implemented approach and widely adopted in CPS intrusion detection [78].

- Support vector machine (SVM)

Support vector machines (SVM) [79][80] are a class of machine learning techniques employed in this work aiming to construct classifiers for the identification of FDI attacks in a normal or anomalous scenario. Besides linear SVM, kernel trick is usually employed when implementing SVM, which transforms the data into a higher dimensional space, where linear methods for classification become applicable. A radial basis function is employed in this work as the SVM's kernel. The RBF kernel is expressed by Eq. (6) [81], where N is the size of training data, γ is a parameter which decides how curvy the classifier's decision boundary could be, and C is a penalty weight. Therefore, when γ is very large, the boundary could be so curvy that the outliers could be classified as labeled but isolated from the correct cluster, and that is when overfitting occurs. When C is large, the penalty term is heavily weighted to avoid misclassification of data and leads to overfitting as well [79]. A comparison study is usually employed to find proper values for γ and C for not losing the generalization properties of the SMV when testing new data.

$$K(x,x') = \exp\left(-\gamma \left\| x - x' \right\|^2\right);$$
$$f(x) = \sum_{i}^{N} \alpha_i y_i K(x_i, x) + b, \text{ where } 0 < \alpha_i < C \tag{6}$$

- Gaussian Processes (GP)

Gaussian processes are a generic nonparametric method in supervised learning, designed to solve regression and probabilistic classification problems. Unlike common regression techniques employing least square to minimize the loss function and output one line or curve to fit the measurements data, the predictions from a GP model take the form of a full predictive distribution. Consequently, the computation of the empirical confidence intervals is appliable, based on which one can decide that if a refit of the prediction in certain regions of interest is necessary. Like SVM, kernel tricks can be implemented in GP as well to capture more sophisticated data variations. However, GP model use the whole samples/features information to perform the prediction, so the computation cost could be a pain for high dimensional spaces. [82]

- Neural Networks

There are all sorts of intelligent-related tasks that can be broken down into layers of abstraction. The activations in one layer determine the activations in the next layer by the mechanism that could conceivably combine raw data into puzzles of patterns, then the puzzles into patterns. [83] Specifically, the question towards neural networks is what parameters should the network have so that the network is sufficiently expressive to potentially capture the patterns in the output layer. This goal is accomplished by assigning a weight to each of the connections between the neurons in the i^{th} layer and the neurons in the $(i+1)^{th}$ layer, and then taking all the activations from the i^{th} layer and compute their weighted sum according to the weights. [84] And learning via neural networks refers to getting the compotation force to find a valid setting for all these weights and biases to solve certain tasks, which could be regression, classification, and so on. Purposefully tweaking these parameters so that the shallower layers pick up on patterns and the deeper layers pick up on signatures.

In recent years, researchers find that building up a little bit of a relationship with what the layers, weights, biases actually mean could make the networks performance closer to their anticipation, which can be attributed to another popular topic, explainable AI (XAI). [85] The interpretation towards the components of the neural network provides a new access to physics insights. And the rise of XAI indicates that there is a trend in fusion of data-driven approaches and physics discipline.

3.5.3 Model-based Techniques

Strictly speaking, a purely model-based approach requires excellent physics model and good specification of the parameter values. However, it would be impractical to harness the complete technical specifications and hidden physical interactions from the first principles since an online monitoring system requires fast model execution and decision-making on the authenticity. In addition, the easy availability of huge online monitoring makes a natural option for using a prudent approach that combines data-driven techniques with extracted information from physics insights. Current studies adopting these hybrid approaches can be broadly categorized into two types: (1) the extracted physics information is represented as features sent to the data-driven models; (2) the physics insights represent in the form of regularizations, like the definition of loss function or

constraints of the data-driven methods. The approaches of the second type are usually used in case of availability of no or less labeled data or insufficient data [45]. But for most monitoring system, the huge amount of genuine historical operation data is available, representing that the availability of datasets labelled as “Genuine”. In addition, the goal of data science and the related subjects aim to subtract as much information from data as possible. Even though sufficient pristine operating data and the corresponding labels could serve as sources of information, the raw representations are not amendable to learning process. Hence, the first choice for evolving physics insights in the control system would be deriving features from raw data.

While investigating the methods and metrics used to evaluate the authenticity of system state and the measurements of process variables, features selection and extraction is most significant procedure throughout the whole process of the algorithm implementation. Whichever the algorithm is employed to recognize patterns in the process variables or identify the correlation between variables, essentially this process aims to subtract information from raw data/ measurements by building derived values, which is denoted as features. The features will facilitate the subsequent learning or modeling steps thereafter; this process is called feature extraction. However, for a complete control system monitoring numerous interrelated process variables, some of the derived features are suspected to be redundant. Inclusion of the full space of features in learning process will lead to impractical computation cost and performance degradation, which are often referred to as curse of dimensionality. Feature selection is adopted to address this issue, usually employing reduced-order modelling (ROM) techniques to determine a subset of the initial feature space, which are expected to contain significant information from the input data. Both feature extraction and selection constitute the contents of feature engineering, shown as the second step in Figure 3.3 [86]. For the majority learning process, raw data are processed via statistical methods or derived from a physics-informed perspective to generate features; then one or more learning algorithms are adopted to train the predictive model for CPS; after evaluation of the trained predictive model, the CPS can make inference of the process variables, and the measurements authentication is fulfilled via comparison between the measurement and the inference. After the whole detection process, the measurements are collected to expand the historical data for updating the predictive model. As most of the monitoring systems generate temporal data, the following background focus on the feature engineering of time series. Based on

the nature of the features and the following employed machine learning methods for FDI detection, the features can be categorized into three different types.

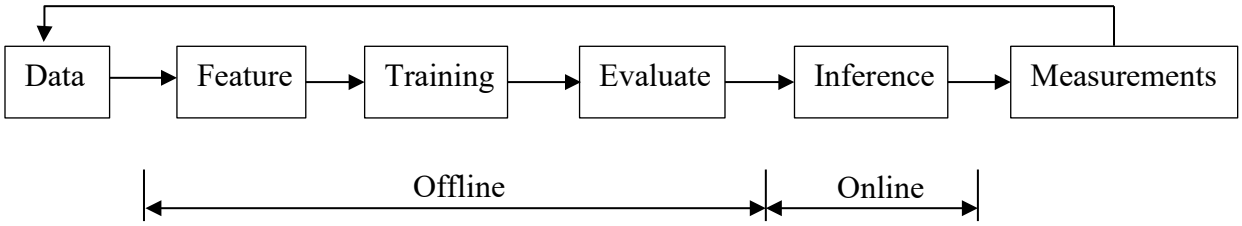

Figure 3.3 Machine learning Execution Flow [86]

3.6 Feature Engineering

There is a vast literature of time-series analysis methods for characterizing time-series properties that can be leveraged to extract interpretable features from a time series. This section describes three types of the most commonly used feature-engineering methods for typical time series. Starting with the simplest time-series metrics, progressively more complicated and sophisticated approaches are described thereafter.

3.6.1 Time domain

Ignoring the timestamps, analyzing the distribution of the time series usually yields informative features for classification, regression, or forecasting. Four simple metrics that simply extract statistical characteristics from the marginal distribution of the time series observations are listed below: [87]

- Average — The median or mean of the time series can uncover the trends in the average value of a time series.
- Variability — The time-series measurements of the spread of a distribution, e.g., standard deviation, interquartile range, or median absolute deviation, can uncover the trends in the spread of the observations.
- Outliers —In many circumstances, such as predicting component failure or process line disruptions, time-series observations that locate many standard deviations away from the

mean value or outside the range of the distribution contain predictive information, e.g., revealing potential anomalies in the observations.

- Distribution — The distribution contains predictive information, including the higher-order characteristics of the distribution of a time series (e.g., skewness or kurtosis), or proceeding with a statistical test for a specific distribution (e.g., Gaussian or uniform) In addition, a more sophisticated set of features that computes windowed statistics entails calculating these statistical properties within a specified time window or a series of time windows. For example, the statistical features generated within a week of measurements may contain predictive information. As a result, windowed discrepancies/similarities can also be determined, which could be employed to differentiate/predict the time series from one time-window to the next.

Next, more-sophisticated time-series feature extraction technique is proposed, matrix profile [88], which measure the similarity of a time series with a lagged version of itself. Specifically, the similarity characteristic of a time series compares the original time series to the time series that has been shifted to the left by one time lag. While this comparison proceeds along the whole time series, one can identify the periodicity and other statistical structure in the time series. Different from the statistical features that are usually calculated as a fixed value or a vector if given a certain series of measurements, while matrix profile engages its greatness in versatility, generality, simplicity, and scalability and refresh time series data. Particularly, it has been broadly applied on time series pattern recognition, clustering, density estimation, shapelet discovery/classification, time series joins, etc. Taking shapelet discovery as an example, shapelets are time series subsequences which are maximal representative of a class. Due to its fast and interpretable classification decisions in a large variety of domains, the shapelet discovery has broad application. Based on its application, the identified shapelets can be denoted as ‘discord’ in anomaly detection, or as ‘motif’ for identifying recurring patterns. In the context of nuclear engineering, the pressure variation at reactor core under the LOCA scenario will be different from the one under normal operation, which can be identified as different shapelets. Then a library of shapelets can be built to include all possible shapelets under various operating conditions, which can be employed for classification of online monitoring data.

3.6.2 Domain Transformation — Spectral Domain

Besides the analysis in time domain, analysis in spectral domain provides another perspective to handle the intractable data in time domain, including wavelet transform, Wigner distribution function, Fourier transform and so on, among which Fourier analysis is one of the most commonly used methods for time-series feature engineering. Fourier analysis aims to decompose a continuous function into a sum of sine and cosine functions on a range of frequencies, existing in many real-world datasets. This decomposition enables quick identification of periodic structure in the function. In time series feature engineering, the discrete data could be expressed by a continuous function, but in most cases, this decomposition is achieved by discrete Fourier transform. The discrete Fourier transform [89] is able to decompose a time series into its spectral components with the corresponding frequency information. And these components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase. Similar approaches that convert the 1D representations into 2D time frequency domain, contains wavelet transform [90], empirical mode decomposition [91] and the advanced variations of these transformations [92].

3.6.3 Correlation/ Dependence Domain

Unlike the feature extraction methods leading to unique/fixed quantities/values mentioned in the previous sections, correlation features measure the statistical correlation of a time series itself (autocorrelation) or among different observables/variables (cross-correlation). For example, a time series' one-autocorrelation feature correlates the original time series with the same time series shifted over by one-time lag to the periodicity and other statistical structure in the time series can be captured by shifting the time series in this manner. Expanding this idea for multivariate problems, one also can look into the dependencies among various variables, and violation of these dependencies can be applied to FDI detections. Taking principal components analysis (PCA) [93] as an example, the number of principal components is usually settled by the decay of singular value spectrum, a restricted cumulative variance, or a user-defined error applied on the reconstructed matrix to restrict the maximum discrepancy.

Feature extraction methods in both time and spectral domains treat the time series of a process variable separately, however, different process variables could be highly correlated, for example, the temperature and pressure at the secondary side of the steam generator. From the perspective of

information amount, the information among process variables is not fully extracted. Several methods are developed to construct features from the variation and distribution of the data, as well as the correlations between different variables.

3.6.3.1 Principal Components Analysis (PCA)

PCA was first proposed by [94] , which tries to find the orthogonal directions representing the variation of the data. Singular value decomposition (SVD), which produces a set of singular vectors and singular values, with the singular values being scalar quantities ordered from high to low, is usually adopted to implement PCA. One can show that the variation in a given variable can be described by a linear combination of the first few singular vectors. In most application of SVD, the singular value spectrum shows a quick decay of its value, the implication is that one needs only few singular vectors to describe the variations for the responses of interest. Mathematically, this is described in Eq. (7) , where $X \in \mathbb{R}^{m \times n}$ represents the normalized raw training data; U represents the orthogonal directions of data variations, which are the eigenvectors of the XX^T, the covariance matrix; Σ represents the diagonal matrix storing descending singular values, and V^T represents the eigenvectors of the $X^T X$. The projections of data on the dominant directions is $U^T X$, which are named as principal components [95][93].

$$\begin{aligned} &\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \\ &\mathbf{U} = [u_1, u_2, u_3 \cdots u_r], \\ &\boldsymbol{\Sigma} = diag[s_1, s_2, \cdots s_r], \\ &\mathbf{V}^T = [v_1^T, v_2^T, \cdots v_r^T]; \end{aligned} \tag{7}$$

3.6.3.2 Kernel PCA

While PCA tries to find the linear subspace to represent the pattern of data, kernel PCA searches for the nonlinear subspace of data. [96] With user-defined nonlinear kernels, the KPCA maps original data onto a higher dimensional space, $x \rightarrow \phi(x)$, then the corresponding kernel matrix is constructed as $K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$, which replaces $x_1^T x_2$ to implement the kernel trick. [97] After applying the kernel trick, SVD is applied on the kernel matrix $K = USV^T$.

Specifically, if the kernel is an identity matrix, i.e., $\phi(x) = x$, the KPCA will reduce to PCA. Most popular kernels include polynomial kernel, radial basis function kernel and so on. Broadly, the construction of kernel matrix encompasses many linear algebra transformations. For example, metric multidimensional scaling (metric MDS), also denoted as Principal Coordinate Analysis (PCoA), is a nonlinear feature extraction method, whose kernel matrix is obtained by $K = -\frac{1}{2}HD^{X}H$. Here, $H \coloneqq I - \frac{1}{n}11^{T}$, is the centering matrix and D^{X} is based on Euclidean distance. Detailed calculation procedures can be found in Ref [98]. Similar methods include but are not limited to Isomap, locally linear embedding, Laplacian eigenmap and so on [99].

3.6.3.3 Supervised Principal component Analysis (SPCA)

As stated in 3.6.3.1, PCA is adopted to solve the problem of finding new directions/variables that are uncorrelated linear functions with maximized variance. SPCA is initially proposed to solve the problems of finding the new variables that are most correlated to one or more responses. The responses could be categorical or numerical data. As a generalization of PCA, the construction of PCs in SPCA is same with PCA, but there is one more procedure, the screening of PCs which is based on the covariance between responses and PCs. As one may think intuitively, regression on the first few dominant components would be a natural option, however, this might not always result in the optimal performance. Bair et al suggest four steps to implement this idea. [100] Firstly, with readily identified PCs, one can compute a set of univariate regression for each PC and obtain the corresponding regression coefficients for each PC.

3.6.3.4 Fisher Linear Discriminant Analysis

Fisher linear discriminant analysis (FLDA) is proposed by Ronald A. Fisher. [101] This approach is often used for classification problems, i.e., analyzing labelled data. Similar to PCA, FLDA also calculated the projection of data along identified directions. Yet, rather than maximizing the variation of data, FLDA attempts to maximize the separability among known categories while minimizing the variation within each category. This goal is fulfilled by maximizing the Fisher criterion, which is formulated as below:

$$J(w) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \tag{8}$$

where **w** represents the discriminant vectors, S_B and S_W represent the between-class and within-class scatter matrix respectively whose mathematical expression is shown as below: [102]

$$S_B = \sum_{c}^{C} (\mu_c - \overline{x})(\mu_c - \overline{x})^T$$
$$S_W = \sum_{c}^{C} \sum_{i \in c}^{C} (\mu_c - x_i)(\mu_c - x_i)^T \tag{9}$$

In Eq. (9), $\overline{x}$ refers to the mean of all training samples, μ_c represents the mean of a certain category, *c*. The category set is denoted as *C*. As stated in 3.6.3.1, the U vectors are calculated as eigenvectors of a certain matrix, such as $\mathbf{X}\mathbf{X}^T$ for PCA, here the discriminant vector **w** are calculated as eigenvectors of $S_W^{-1} S_B$.

3.6.3.5 Independent Component Analysis

ICA was initially proposed to solve blind source signal separation problem. As the Different from PCA, which finds the orthogonal directions from the original data, i.e., each pair of PCs has the least covariance, and the significance of these directions are ranked based on the singular values, ICA aims to find independent directions, which are non-orthogonal and unranked, but the combination of each pair of the ICs has the least mutual information. [103] Due to the properties of ICA, ICA can only exploit non-Gaussian data source, since the rotational symmetry of Gaussian data introduces ambiguity in separating the data sources. [104] ICA can be implemented using different methods, i.e., different objective function and optimization algorithm. For example, Hilbert-Schmidt Components Analysis (HSCA) employs Hilbert-Schmidt Independence Criterion (HSIC), a measure of dependence between two random variables, as the metric to identify eigenvectors, **U**, with maximized HSIC, which serves as a set of basis to construct features. The detailed implementation can be found in [105].

3.6.4 Evaluation metrics for FDI Detection

Since FDI detection is usually a bifurcation problem, the datasets used for network anomaly detection for classification are usually labelled as 'no attack' and 'attack' instances. Based on the discussion in the earlier sections, it is indispensable to employ proper metrics for FDI attack countermeasures evaluation. Based on the prediction and the actual label, there are four types of combination, shown in Table 3, denoted as confusion matrix. [106] All correct predictions are located in the diagonal of the Table 3. The cases labelled as 'attack' is denoted as positive, which locate on the first line of the confusion matrix. Therefore, each instance can be represented by the alignment between prediction and actual labels as well as the predicted attack condition. One can easily find the prediction error happens outside the diagonal of the table, and the false negative (FN) instances worth more attention, since there is an attack in these instances while the prediction will not issue an alarm. Based on the confusion matrix, a series of probability can be calculated to evaluate the correctness/performance of binary classification, as listed below.

Table 3 Confusion Matrix

		Actual Labels	
		Attack	No Attack
Predicted Labels	Attack	True positive (TP)	False Positive (FP)
	No Attack	False Negative (FN)	True Negative (TN)

- True Positive Rate (TPR)

TPR, also denoted as sensitivity or recall, represents the probability of the attack detection, expressed in Eq. (10).

$$TPR = \frac{TP}{TP + FN} \tag{10}$$

- True Negative Rate (TNR)

TNR also known as specificity and selectivity, represents the probability of the correctly predicted normal conditions, expressed as:

$$TNR = \frac{TN}{TN + FP} \tag{11}$$

- False Positive Rate (FPR)

FPR represents the probability false alarm, also denoted as type I error in a binary classification problem, expressed in Eq. (12. If all 'no attack' cases are predicted as labelled, the FPR will reach to 0.0 while the TNR will reach to 1.0.

$$FPR = \frac{FP}{TN + FP} \tag{12}$$

- False Negative Rate (FNR)

FNR is also denoted as miss rate or type II error, representing the proportion of positives which yield negative classification prediction, expressed in Eq. (13). Ideally, if all the attack instances are identified, the TPR will reach to 1.0 and the FNR will be 0.

$$FNR = \frac{FN}{TP + FN} \tag{13}$$

- Precision / Positive Predicted Value (PPV)

Precision represents, among all the instances with predicted as 'attack', the portion of the correctly labelled ones, expressed as Eq. (14). Like TNR and TPR, but PPV is defined from a perspective of the creditability of the predictions. PPV reaches its best value at 1.0 and the worst value as 0.0.

$$PPV = \frac{TP}{TP + FP} \tag{14}$$

- Accuracy

Accuracy represents the proportion of correct predictions (both true positives and true negatives) among the total number of examined instances, expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{15}$$

- F_1 score

F_1 score, as known as F-measure, is a measurement that considers both PPV and TPR to compute. The F_1 score can be interpreted as a weighted average of the PPV and TPR, where an F_1 score reaches its best value at 1 and worst value at 0, as expressed in Eq. (16). F_1 score is usually more useful than accuracy, especially for an uneven class distribution. For instance, if the cases labeled as 'attack' occupy a small portion of all cases, e.g., 3%, none of which are identified as 'attack' data, the accuracy will reach 97%, while the corresponding F_1 score will be 0, since there is no identified true positive case.

$$F_1 = 2\,\frac{PPV \times TPR}{PPV + TPR} \tag{16}$$

4 PRELIMINARY STUDY I: LOCS RECOVERY [41]

(A version of this chapter has been previously published in Nuclear Science and Engineering, with DOI: 10.1080/00295639.2020.1840238.)

Each mathematical or physics model consist of different variables and their relationships. These variables contains (1) independent variables ,x, which represent the quantities that can be manipulated in the model or experiments; (2) exogenous variables, known as parameters, which are usually constants showing in the relationship between the independent variables ant the observable variables of the mode; (3) random variables, δ, which represent the uncertainties or noise of the model, usually obeying a certain statistical distribution; (4) dependent variables, also known as response or observable variables, usually denoted as y, which are functions of the above three types of variables. Thereinto, independent and dependent variables are outside the model function box, which are easier for attackers to have access to. But the model parameters are inside the function box, which the attacker cannot directly obtain from operational data. Two preliminary studies stated in Chapter 4 and 5, employ different approaches to indicate:

(1) One can leverage data-driven approach to recover the dominant components (LOCs) solely with operational data. (Chapter 4)
(2) Given an approximate model, one can avail of operational data to learn the model parameters. (Chapter 5)

4.1 Current Methods for LOCs Recovery

As stated in Introduction, the model-based approaches can be either data-driven or physics model driven. The methods developed in the Exploratory study (Chapter 6 and Chapter 7) are physics based, but in principle, one can learn the model on the fly from the operational data. For demonstration, this work employs neural networks to construct surrogate model. The results indicate that solely with data repository, one can capture the general trend of the temporal evolution, which provides a prototype for the construction of plausible attack vectors to circumvent the basic statistical checks.

The specific example employed in this Chapter is the simulation of the protected and unprotected Shut Down Heat Removal Tests (SHRT-17 and SHRT-45R) for the Experimental Breeder Reactor II (EBR-II), modeled using the System Analysis Module (SAM), developed at Argonne National Laboratory for advanced non-LWR safety analysis. The data-driven learning process, the basic idea is to try to regress one set of variables, referred to as responses, to other set of variables, called regressors. To regress, by definition, is to explain the cause of responses. In doing so, the relationship between the regressors and responses, referred to as response surface or a surrogate model, is often based on trial and error until an acceptable surface is identified which minimizes the regression errors. Some of the notable choices for the surrogate model include linear/polynomial functions [107], artificial neural network-based functions [108], Gaussian process models [109], etc. The selection of a certain surrogate model is often guided by the nature of the physics model or phenomena being analyzed, and hence is generally a subjective process. Thus, it is not surprising that multiple surrogates could be developed with essentially similar accuracy. In this study, both linear regression and artificial neural network (ANN) based reduction are employed to construct surrogate models in terms of the active DOFs generated using pattern-based reduction.

4.2 Model Description

This work utilizes SAM transient simulations of the Experimental Breeder Reactor (EBR-II) Shutdown Heat Removal Tests (SHRT) SHRT-17 and SHRT-45R tests, protected (scrammed) and unprotected (unscrammed), respectively, loss of flow tests performed at the facility to characterize and quantify the inherent safety characteristics of the pool-type sodium-cooled fast reactor. In both experiments, transient conditions are initiated by tripping the primary coolant pumps from nominal system states. In SHRT-45R, this results in an initial increase in primary system coolant temperatures. The inherent properties of metal fuel and the core design enable negative reactivity insertion, decrease in reactor power, and a subsequent, unassisted cooldown of the primary system as natural circulation flow patterns begin to develop. In SHRT-17, with the reactor in a scrammed state, the key safety behavior demonstrated included development of natural circulation flow and successful performance of the decay heat removal pathway.

The goal of this work is to explore the reducibility of the SAM code using pattern-based and surrogate-based reduction techniques. The overarching goal here is to determine whether additional reducibility can be incorporated into the SAM physics model, based on the range of its intended application, thereby providing an efficient solver capable of performing computationally intensive analyses such as uncertainty quantification, inference, etc., especially when the number of model parameters is significantly increased. A representative model with 25 input model parameters expected to directly influence key performance metrics (fuel, clad, and coolant temperatures) is employed. the uncertainties of which have been reproduced from Ref. [110] and shown in Table 1. The training snapshot are generated based on 1000 model executions, each randomizing the input parameters within their prior uncertainties. Each execution records the model responses and the associated state. The time-dependent fuel temperature is selected as the state variable, while the peak temperature over the transient time is selected as the model response. The goal is to create a reliable ROM model relating input parameter variations to both the state and response variations over the range of uncertainties for the model parameters. The transient time is selected to be 900 seconds, which corresponds to benign termination of the transient. The details of two models employed are listed in Table 2. More detailed description of both models may be found in Refs [110] [111].

Table 4 Input Parameter Uncertainties for SAM TH Model of EBR-II SHRT Experiments

Input parameter	Uncertainty	Distribution
Initial Power	0.5MW	Normal
Initial Pump 1 Head	0.01 bar	Normal
Initial Pump 2 Head	0.01 bar	Normal
IHX Secondary Inlet Temperature	0.5 K	Normal
IHX Secondary Inlet Temperature	0.6%	Normal
Peak Channel Flow Area	1.0%	Uniform
Peak Channel Hydraulic Diameter	1.0%	Uniform
Fuel Pin Gap Size	1.0%	Uniform
System Heat Transfer Coefficient	-6.5% +32.8%	Uniform
Fuel Channel Heat Transfer Coefficient	30%	Uniform
IHX Primary Heat Transfer Coefficient	30%	Uniform
System Wall Friction	10%	Uniform
Channel Wall Friction	26%	Uniform
IHX Primary Wall Friction	26%	Uniform
Gap Thermal Conductivity	10%	Normal
Fuel Thermal Conductivity	6%	Normal
Cladding Thermal Conductivity	5%	Normal
Gap Heat Capacity	4%	Normal
Fuel Heat Capacity	5%	Normal
Cladding Heat Capacity	2%	Normal
Coolant Density	0.4%	Normal
Coolant Compressibility	2%	Normal
Coolant Heat Capacity	3%	Normal
Coolant Viscosity	5%	Normal
Coolant Thermal Conductivity	15%	Normal

Table 5 Comparison of Simulation Models

	SHRT-17 (Protected loss-of-flow transient)	SHRT-45R (Unprotected loss-of-flow transient)
Incidents description	From full power and flow conditions, all coolant pumps are tripped to simulate a loss-of-flow accident	
Control rod	Control rod insertion was disabled	Control rod insertion was disabled
Power transient	Immediate power scram	Solely rely on reactivity feedback
Model difference	Lumped core model	Detailed core model
Initial Power	57.3 MW	59.9 MW
Initial Pump 1 Head	2.930 bar	2.890 bar
Initial Pump 2 Head	2.929 bar	2.890 bar
IHX secondary inlet temperature	574 K	560 K
IHX secondary inlet flow	-0.83 m/s	-0.81 m/s

4.3 Obtain LOCs – using Neural Network

4.3.1 Data-Driven surrogate modeling for SHRT-17

The first test case is protected loss of flow test SHRT-17. The second test case is unprotected loss of flow test SHRT-45R. In both cases, the temporal profiles of the fuel temperature, cladding, and coolant are selected as responses at spatial locations where the peak values are expected.

The fuel, clad, and coolant temperatures versus time are used as the state function representing the snapshots collected from each model execution. Employing SVD, the active subspace, i.e. LOCs, can be identified. Figure 4.1 plots the error bound obtained as a function of the size of the active subspace for each the fuel, clad, and coolant temperatures. Assuming an acceptable tolerance of 2% of the mean temperature value, a rank of 5 is achieved, which is two orders of magnitude smaller than the number of time steps employed in SAM simulation.

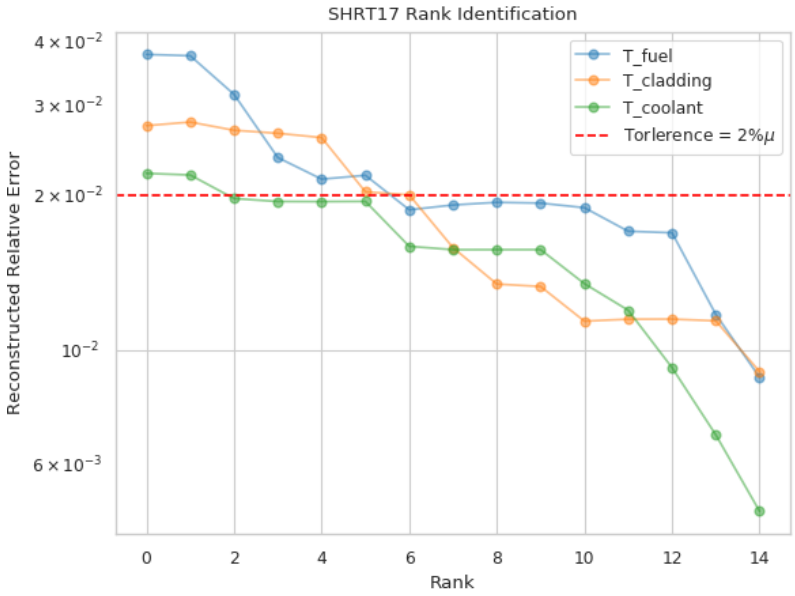

Figure 4.1 Reduction of Temperature Temporal Evolution

Taking the fuel temperature as an example, Figure 4.2 shows all the snapshots, represented by the grey cloud in Figure 4.2 (a), surrounding the mean value in red, and the first five active DOFs, i.e., $\varphi_i(t)$, representing the first five singular vectors from the SVD decomposition as shown in Figure 4.2 (b).

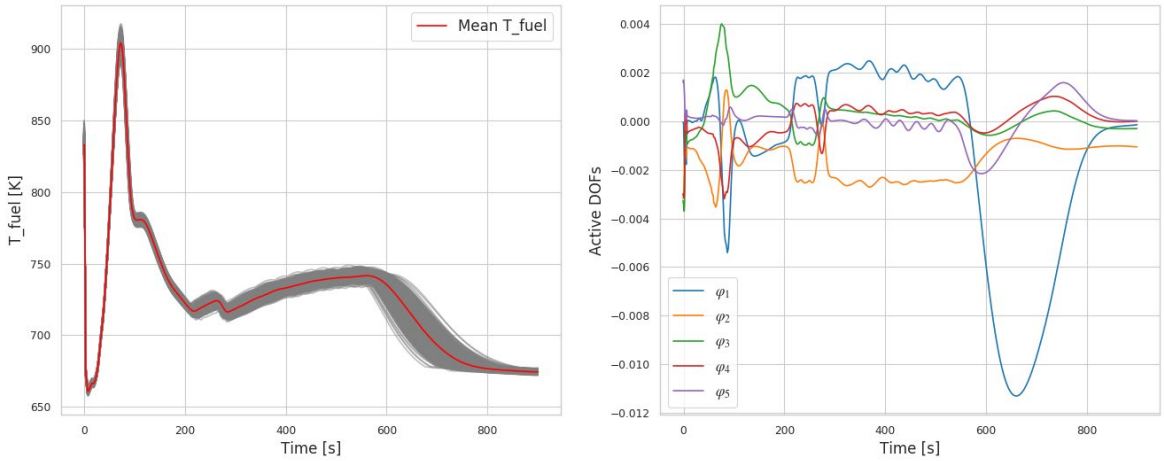

Figure 4.2 (a) Transient Fuel Temperature Snapshots (b) Active DOFs along Temporal Axis of SHRT-17

The implication is that each of the training snapshots in the left figure can be approximated as linear combination of the five basis functions in the right figure per Eq. (1) such that the error calculated by Eq. (3) is guaranteed to be less than the user-selected tolerance per Eq. (4).

Next, a regression-based reduction is applied, wherein each of the state's active DOF is functionalized in terms of the input model parameters. After the regression is completed, the fitted active DOFs are combined back to estimate the state function. Mathematically, this may be described as follows. Let $\phi(t)$ represent the state, and the corresponding active DOFs be given by: $\alpha_i = \varphi_i(t)^T \phi(t)$, representing the inner product between each of the active DOFs basis functions $\varphi_i(t)$ and the state. A regression-based model is employed to approximate α_i as a function of the model parameters x, i.e., $\alpha_i^{reg}(x)$. This is achieved based on the training data $\alpha_i^{trn}(x_j)$, j=1,….,N_{trn}, where x_j represents the j^{th} sample for the model parameters, and N the total number of training samples. The ^{reg} superscript denotes the regression-estimated active DOF, and ^{trn} superscript denotes the value calculated by the SAM code. After the regression is completed, the state is estimated by:

$$\phi(t) = \sum_{i=1}^{r} \alpha_i^{reg}(x)\varphi_i(t) \tag{17}$$

where r denotes the number of active DOFs.

Two different regression-based reductions are employed. First, an ANN-based regression is applied using a Python-based package, Keras [112], with two hidden layers. The activation function for input and convolution layers is Relu function and for output layer is softmax function. The mean squared error is selected for the loss function, and an optimizer using stochastic gradient descent algorithm is employed to minimize the loss. The 1000 samples are divided into 2 parts: N_{trn}=700 for training and N_{tst}=300 for testing. The layout of the ANN is shown as an architecture in Figure 4.3, whose layer is represented as a table with the number of activation function for both of input and output layer and the data flows as the direction of the arrows The comparison between the predictions to the original temporal temperature profile is employed to evaluate the adequacy of the ANN model as shown in Figure 4.4. The testing data are employed to validate the trained model, i.e., evaluate the adequacy of the ANN models by comparing their predictions to the original model prediction, as shown in Figure 4.4.

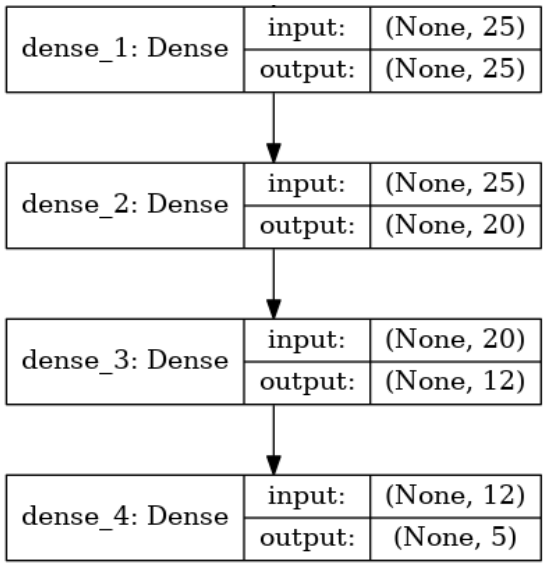

Figure 4.3 Neural Network Layout

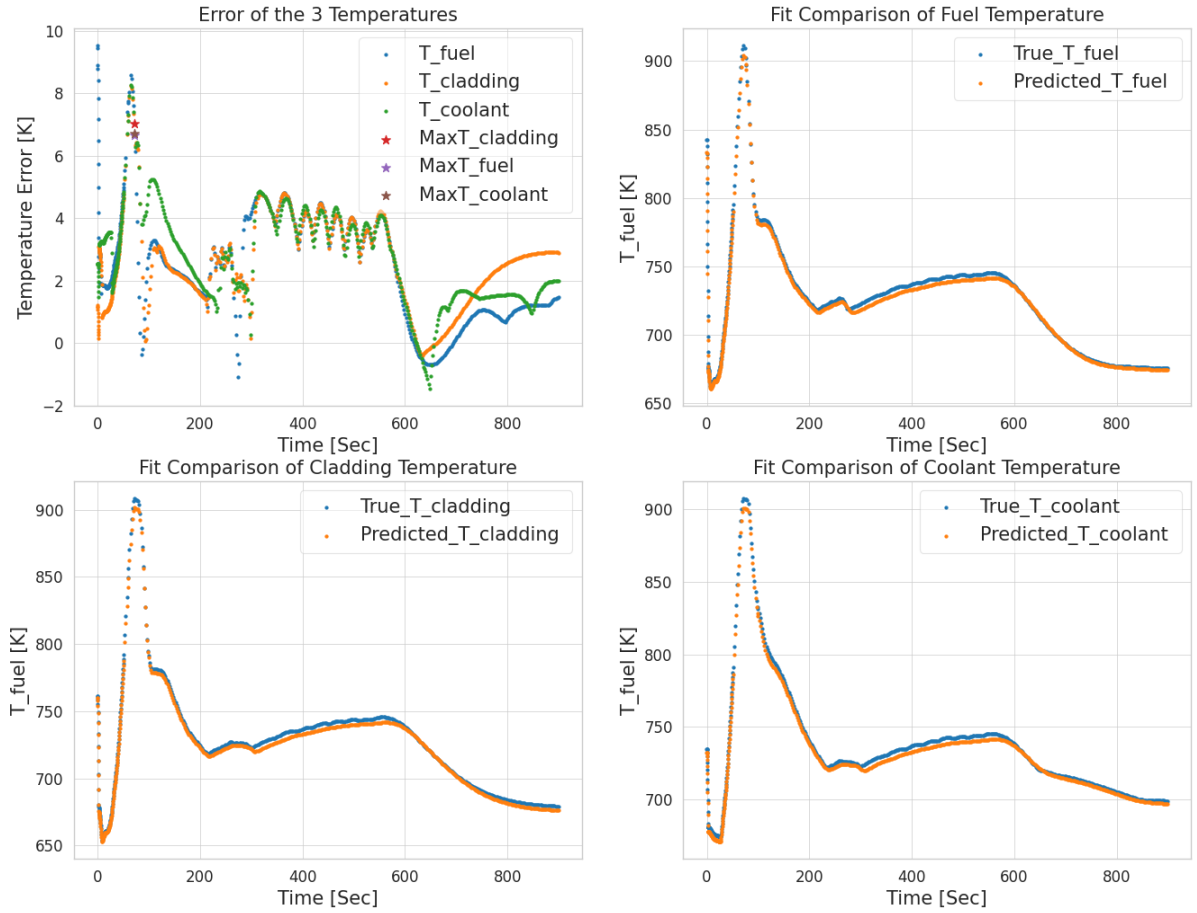

Figure 4.4 ANN Performance for Coolant Temperature – SHRT-17

Figure 4.5 condenses the above results into one figure, plotting the frequency of the errors recorded for all three responses over all 300 testing samples and all time steps. Figure 4.6 displays the results in a similar manner but only for the time steps where the peak temperature occurs, since the peak temperature represents the basis for thermal margin specification. Results indicate that the errors are small enough and below the acceptable safety margin, ranging from 100K to 200K for fuel, cladding and coolant, as specified in Ref. [110].

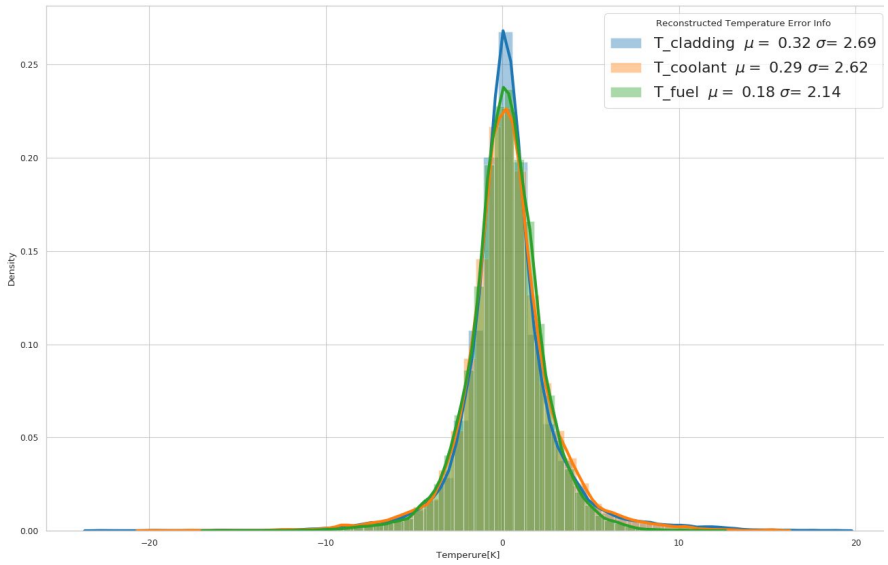

Figure 4.5 Overall Surrogate Modeling Error Distribution of SHRT-17 Case Study

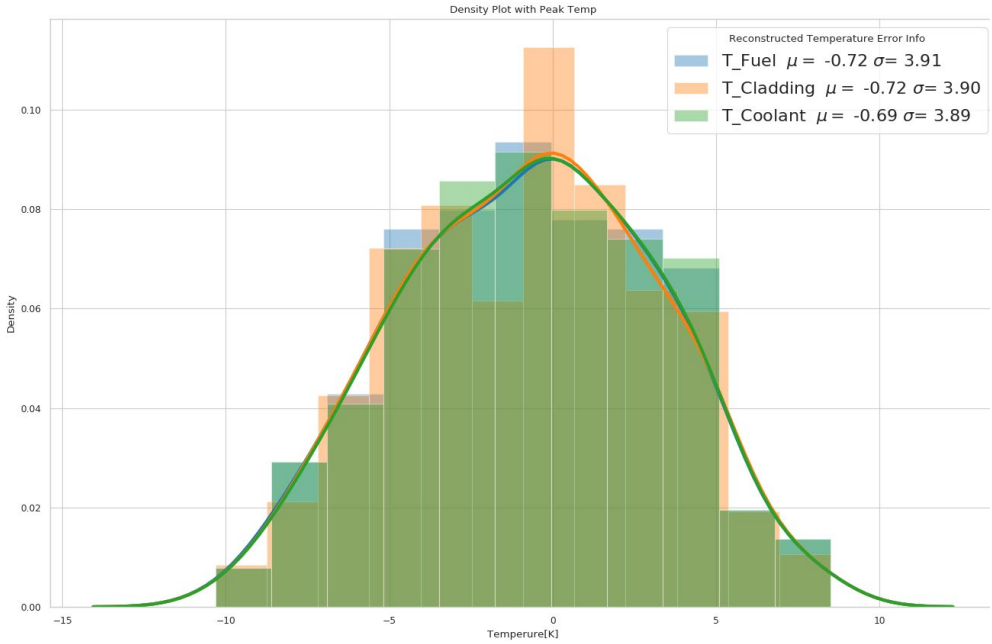

Figure 4.6 Surrogate modeling Error of Peak Temperature in SHRT-17 Case Study

If the reduction errors are not deemed acceptable, one could gain more insight by plotting the reduction errors versus time for all the validation snapshots as illustrated in Figure 4.7 through Figure 4.9 which respectively plot the errors for the fuel, clad, and coolant. The z-axis represents the discrepancy between predicted temperature and testing data in Kelvin units. For example, for the fuel temperature, most of the high errors are concentrated between 550 to 640 seconds.

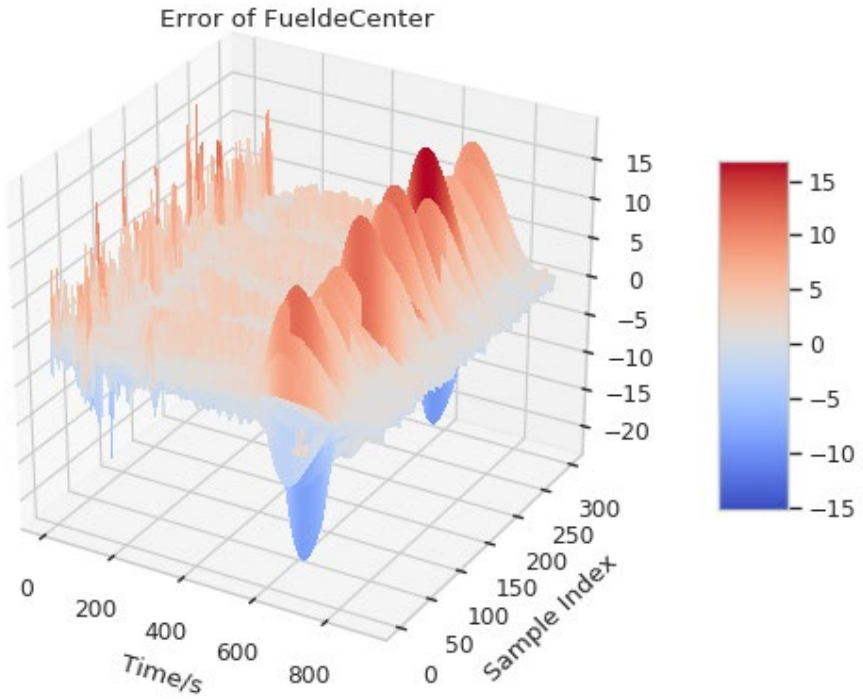

Figure 4.7 Reduction Errors of the Fuel Temperature

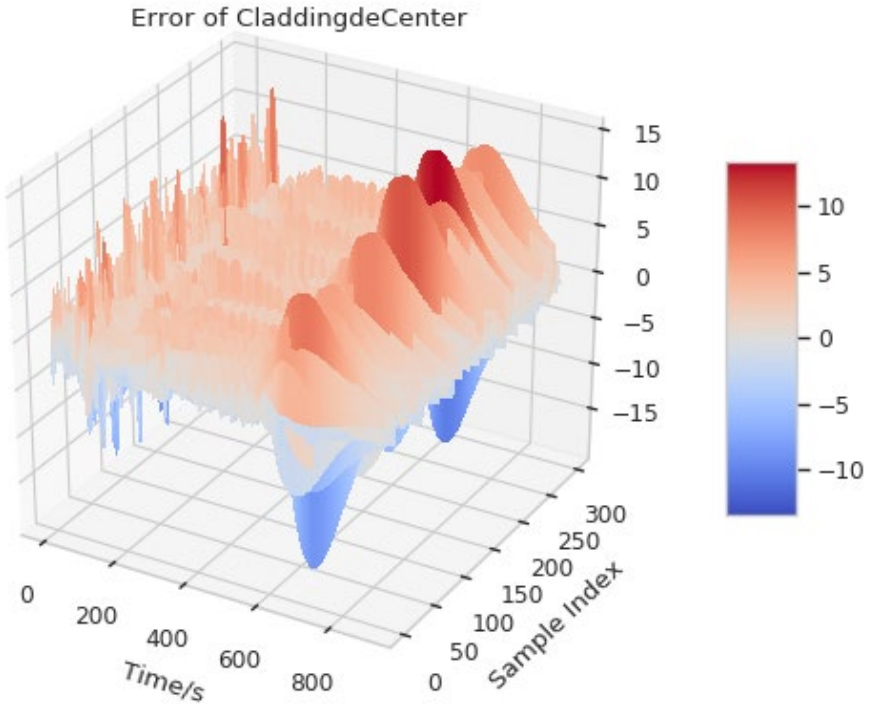

Figure 4.8 Reduction Errors of the Cladding Temperature

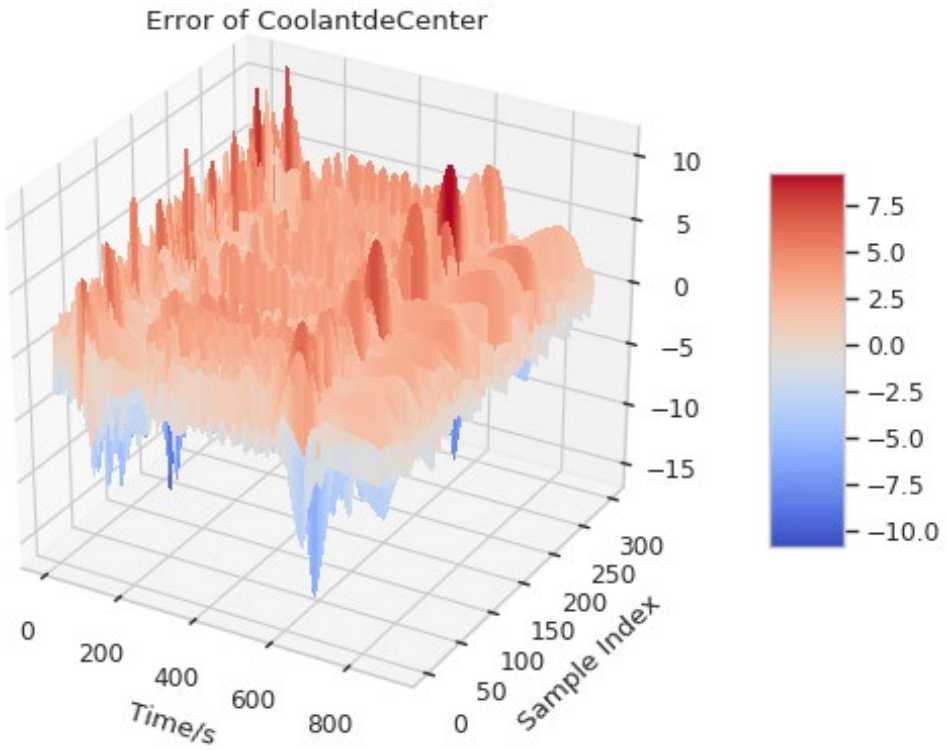

Figure 4.9 Reduction Errors of the Coolant Temperature

4.3.2 Data-Driven surrogate modeling for SHRT-45R

The above results are repeated for the SHRT-45R test, i.e., the unprotected loss of fluid incident. Figure 4.10 shows the transient fuel temperature profile of the SHRT-45R incident. Compared to the transient fuel temperature in SHRT-17 case in Figure 4.2, the peak temperature is 50K higher, the transient variation is more significant, and it takes more time to reach a steady state, since the disabled control rods lead to a dramatic temperature rise and a rapid decay after the peak due to the large feedback effects.

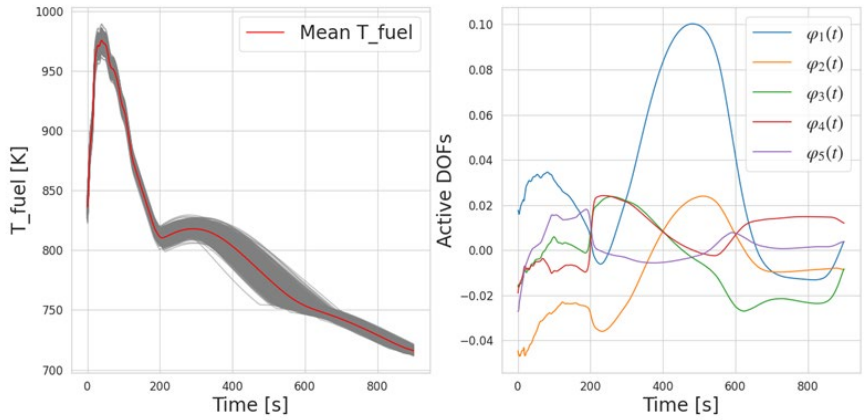

Figure 4.10 (a)Transient Fuel Temperature Snapshots (b)Active DOFs of Fuel Temperature of SHRT-45R

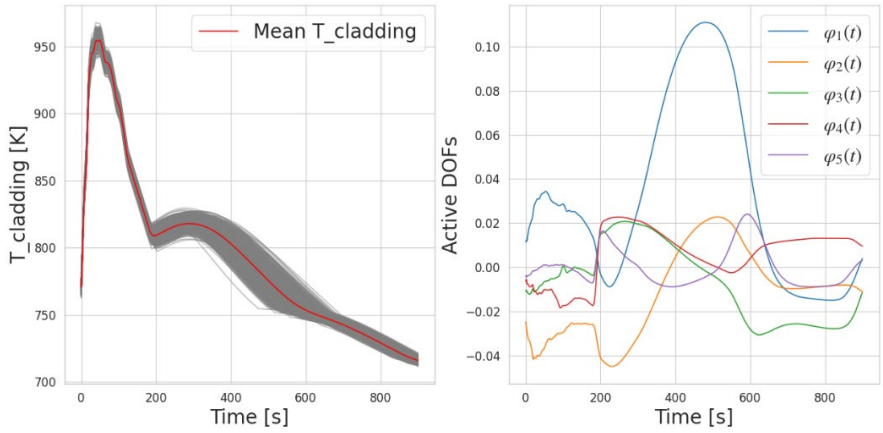

Figure 4.11 (a)Transient Cladding Temperature Snapshots (b)Active DOFs of Cladding Temperature of SHRT-45R

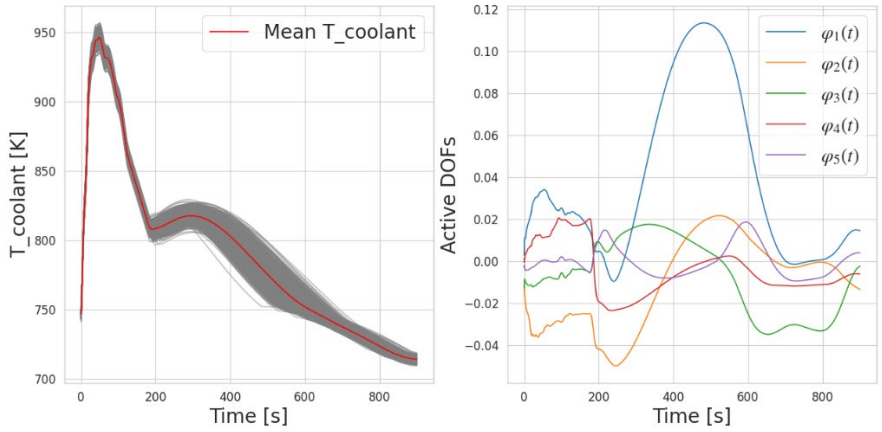

Figure 4.12 (a)Transient Coolant Temperature Snapshots (b)Active DOFs of Coolant Temperature of SHRT-45R

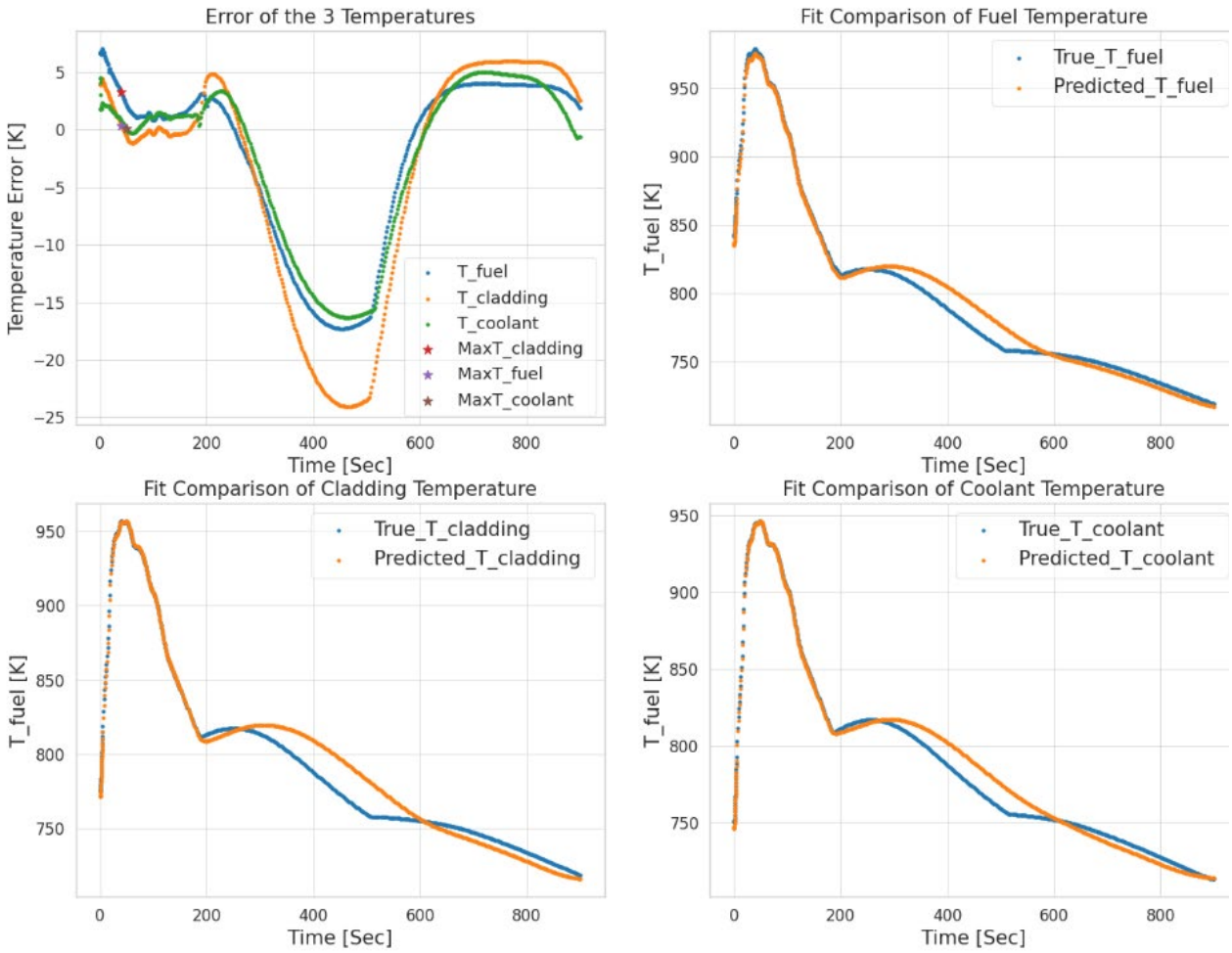

Figure 4.13 ANN Performance of Fuel Temperature – SHRT-45R

Figure 4.14 -Figure 4.16 show higher errors as compared to those of the SHRT-17 benchmark, which is within the expectation because the SHRT-17 benchmark is modeled by lumped approach while the SHRT-45R is simulated with more details in core area. Moreover, for most simulation runs, the largest error happens during 580~660 seconds, the natural circulation transition period. Due to the complicated physics in this region, it is difficult for the straightforward application of neural network to capture this complexity without additional customization based on domain knowledge.

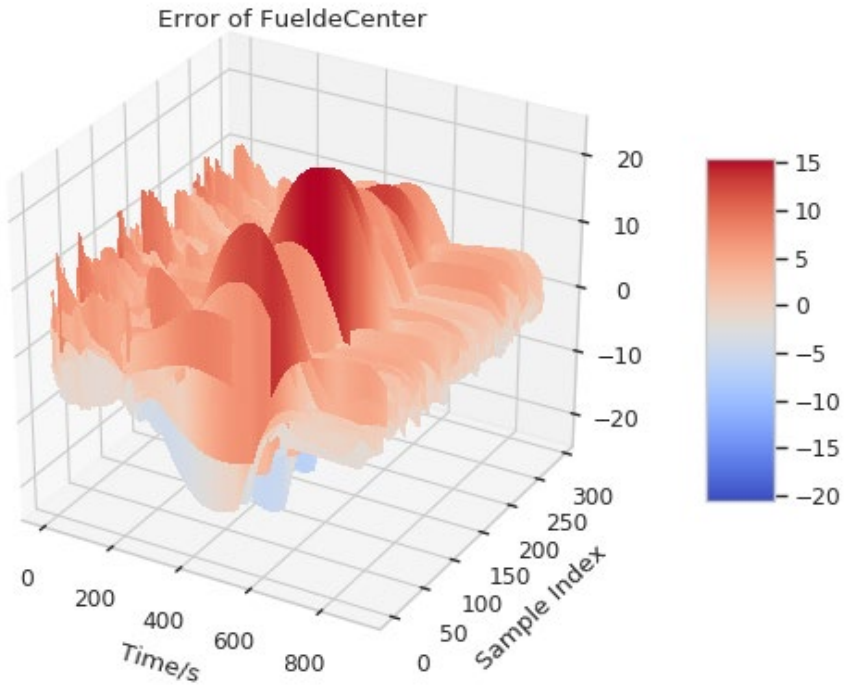

Figure 4.14 Error Profile of Fuel Temperature in SHRT-45R from Surrogate Modeling

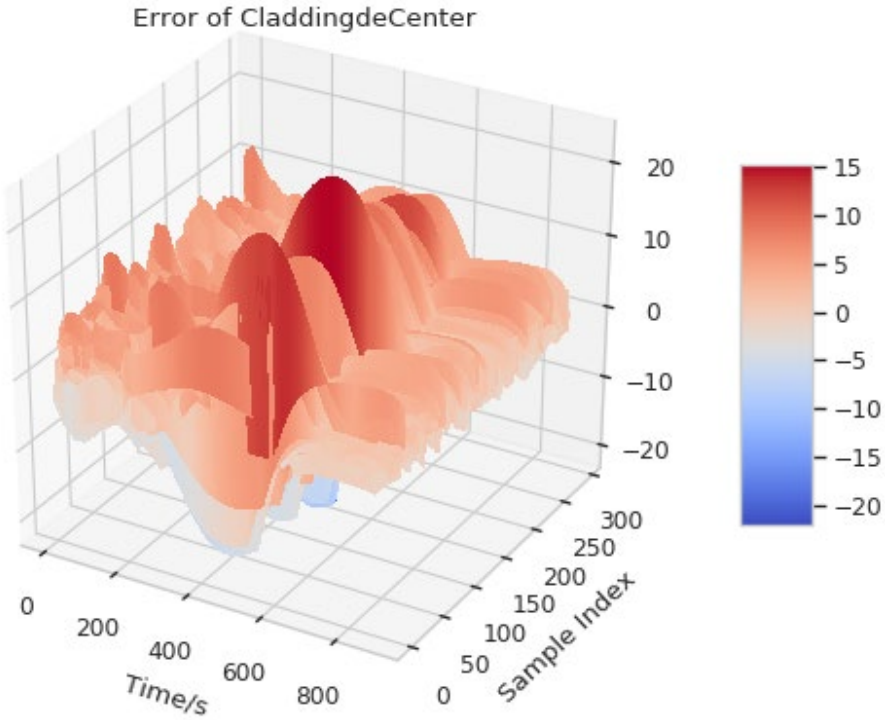

Figure 4.15 Error Profile of Cladding Temperature in SHRT-45R from Surrogate Modeling

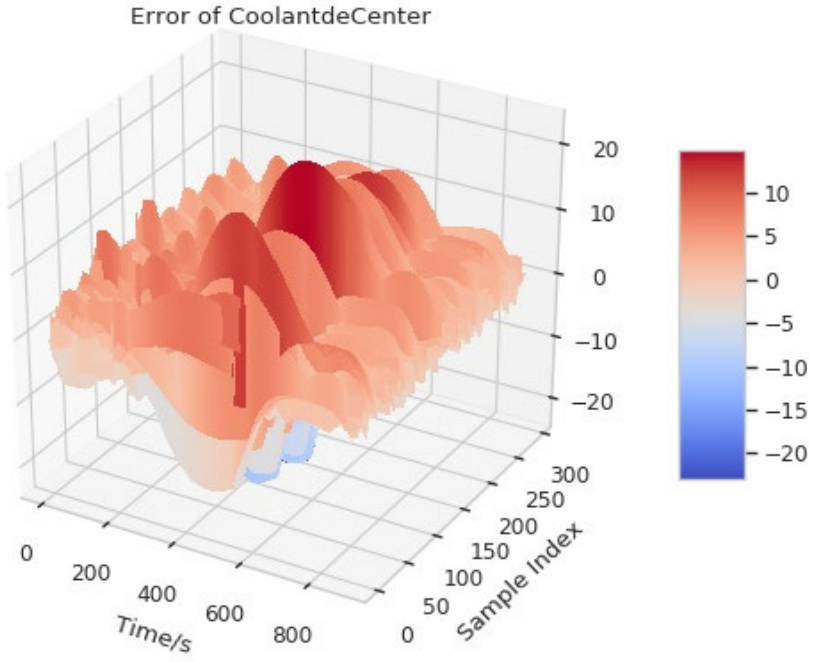

Figure 4.16 Error Profile of Coolant Temperature in SHRT-45R from Surrogate Modeling

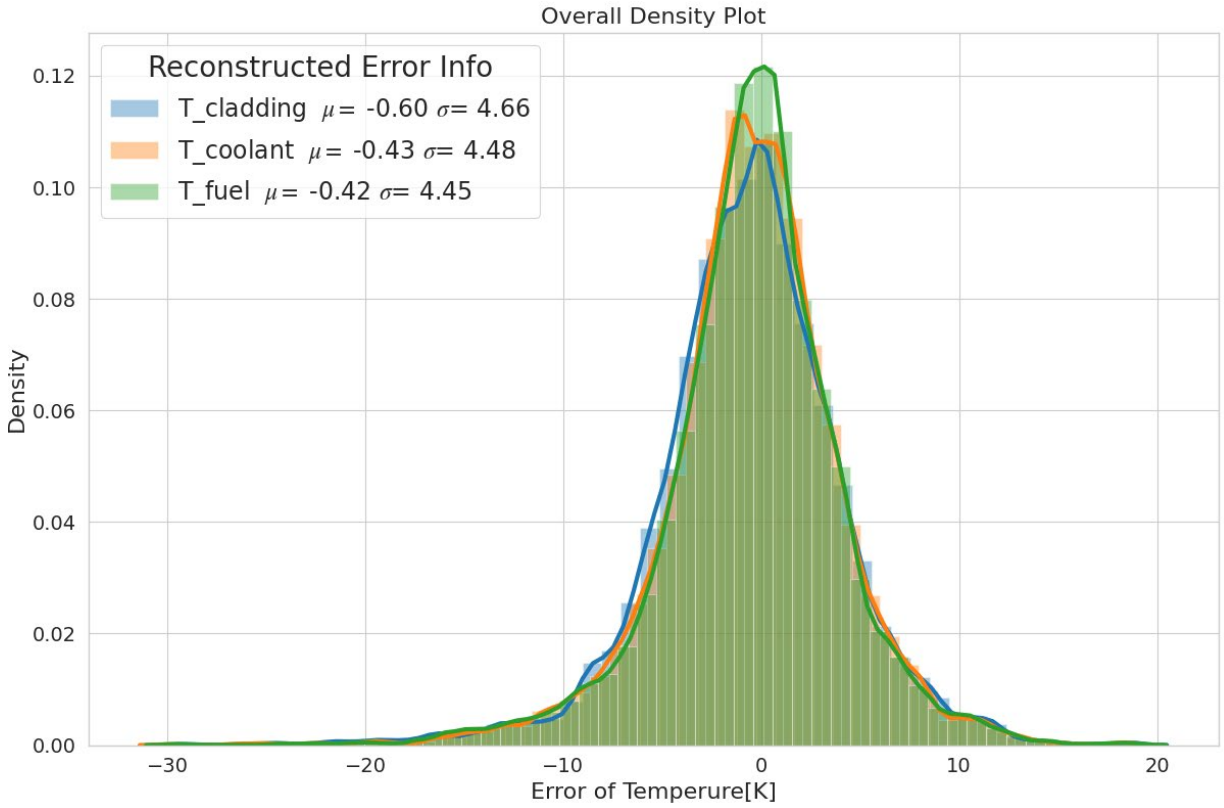

Figure 4.17 Overall Surrogate Modeling Error Distribution of SHRT-45R Case Study

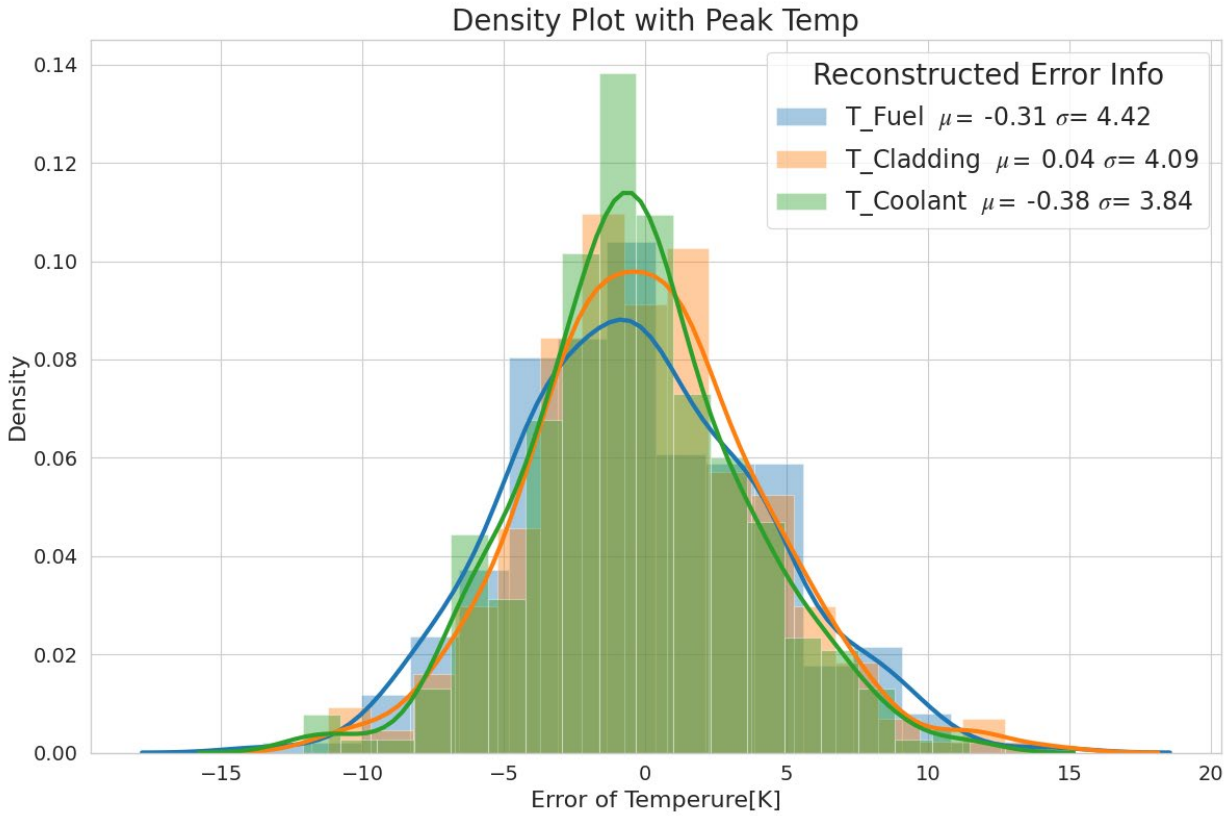

Figure 4.18 Surrogate Modeling Error Distribution of the Peak Temperature in SHRT-45R Case Study

4.4 Results Summary

The surrogate modeling for both benchmarks of EBR-II has relatively small error, whose statistical information can be found in Figure 4.18. The recovery of the temporal evolution of the temperature indicates that without any insights/ knowledge of a physics model, neural network can mimic the physics model behavior to a great extent. In other words, as long as the attackers have access to operational data, they can leverage a data-driven approach to learn a general profile of the reactor model, which represents the LOCs of the model. With learned LOCs of different responses/process variables, the attackers would be capable of launching a coordinate cyber-attack that intrudes false data in different locations or for different responses.

5 PRELIMINARY STUDY II: MODEL RECOVERY [113]

(A version of this chapter has been previously published in Nuclear Technology, with DOI: 10.1080/00295450.2019.1626170.)

As stated in Chapter 1, attackers are able to learn the LOCs of a physics model with access to operational/historical data. To this extend, another question stated in the Introduction that whether the

For the first question stated in the introduction, neural networks, including single and deep-layer networks, will be used to build predictive models for system behavior using both passive and active monitoring. Active monitoring means that the attackers can inject small perturbations to the commands sent to the actuators to induce small variations in the system state that can be used to improve the attacker self-learning process. Active monitoring is a strategy employed by attackers when they have little knowledge about the system dynamics, or the defense measures in place. To answer the second question, a simplified physics model based on public information is employed with some undetermined parameters, whose true values are assumed hidden from the attacker, e.g., representing proprietary design details. Inference techniques will be used to estimate the true values of those parameters via an objective-function-guided minimization of the discrepancies between monitored and predicted variables.

5.1 Model Description

A point kinetic model, based on the Iranian Bushehr nuclear reactor [114], is employed as research object. The model is based on four differential equations which follow the evolution of reactor flux or power, delayed neutron precursors, Iodine, and Xenon concentrations. Xenon decays radioactively and is produced from both fission and the decay of Iodine, which is generated from fission.

$$\frac{dP}{dt} = \frac{\rho_{net} - \beta_{eff}}{\Lambda} P + \lambda_{eff} C \tag{18}$$

$$\frac{dC}{dt} = \frac{\beta_{eff}}{\Lambda} P - \lambda_{eff} C \tag{19}$$

$$\frac{dI}{dt} = \gamma_I \Sigma_F P - \lambda_I I \tag{20}$$

$$\frac{dXe}{dt} = \gamma_{Xe} \Sigma_F P + \lambda_I I - \lambda_{Xe} Xe - \bar{\sigma}_{Xe} XeP \tag{21}$$

The P(t) is the reactor power, which is assumed to represent the response measured by the sensors. In (1), ρ_{net} denotes the net reactivity, which demonstrates how neutron source and feedback effects work on the system, where $\bar{\sigma}_{Xe}$ is an effective value for the Xenon absorption cross section, expressed in (6).

$$\rho_{net} = \rho_{ext} - \alpha_P [P(t) - P_0] - \frac{\bar{\sigma}_{Xe}}{\upsilon \Sigma_F} [Xe(t) - Xe_0] \tag{22}$$

$$\bar{\sigma}_{Xe} = \frac{\sigma_{Xe}}{\Sigma_F E_F V} \tag{23}$$

The function ρ_{net} may be thought of as the forcing function that is manipulated by the controller, the first term can be adjusted via physical changes to the reactor, i.e., moving the control rods, increasing flow rate, etc., and the other two terms are natural feedback, and hence cannot be controlled. In our notations, ρ_{net} represents u_n. All design parameters in this point kinetic model are listed in Table 6 [114].

Table 6. Designed Parameters in Point Kinetic Model

Symbol	QUANTITY	Value
$P(t)$	Temporal core power	P_0 =3000MW
$C(t)$	Temporal precursor concentration	
$I(t)$	Temporal Iodine concentration	
$Xe(t)$	Temporal Xenon concentration	
ρ_{ext}	external reactivity injected into the core	
ρ_{net}	net reactivity of the core	
α_p	power coefficient of reactivity (temperature dependent feedback)	0.48×10^{-11} W^{-1}
β_{eff}	effective delayed neutron fraction	700×10^{-5}
λ_{eff}	effective precursor decay constant	7.841×10^{-2} s^{-1}
Λ	neutron mean generation time in the core	32×10^{-6} s
υ	average number of neutrons produced by fission	2.45
Σ_F	effective one group fission cross section for the core	0.77×10^{-2}
γ_I	fission yield for Iodine	6.386×10^{-2}
λ_I	Iodine decay constant	2.875×10^{-5} s^{-1}
γ_{Xe}	fission yield for Xenon	0.228×10^{-2}
λ_{Xe}	Xenon decay constant	2.092×10^{-5} s^{-1}
σ_{Xe}	microscopic neutron capture cross section for Xenon	2.7×10^{-18} cm^2
E_F	Energy released per fission	320×10^{-13} J
V	Core volume	27.8 m^3

5.2 Physics-based model

This section discusses the basic physics model employed by the attacker, as well as the mathematical procedure employed to maximize its predictability against the monitored sensors data.

In an adversarial setting, this work assumes the attacker has the equations described in section 3.1.1, but does not know the exact values of three parameters, namely: the power feedback coefficient α_p, the microscopic neutron capture cross section of Xenon σ_{Xe}, and the fission cross section Σ_F. By relying on the physics model, the attacker can approximate the effect of each parameter on the system state via a parametric study. During an initial lie-in-wait period, the attacker can insert small perturbations to the commands, u_n, or by varying the recorded sensor signals, y_n, can excite state variations that can be used to estimate the true values for the unknown parameters using inference techniques. For sake of demonstration, Figure 5.1, Figure 5.2 and Figure 5.3 show the effect of the variations of each of these three parameters on the reactor power,

found to impact the oscillatory behavior characteristic, in terms of amplitude, phase shift, and damping speed. The power coefficient of reactivity affects the natural feedback from fuel temperature and coolant temperature, and hence is expected to impact the power amplitude. The other two coefficients perturb the balance between the rate of Xenon production and destruction and hence are expected to have more impact on the oscillatory behavior in terms of the phase shift and damping speed.

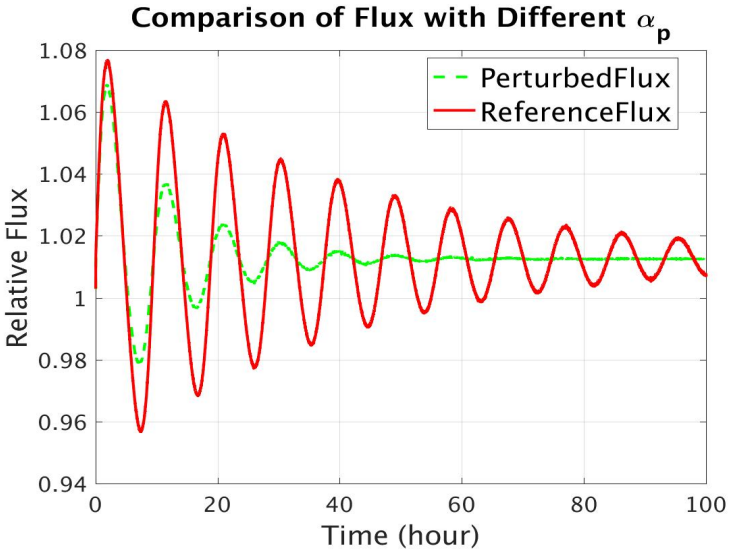

Figure 5.1 Power Sensitivity due to Parameter α_P

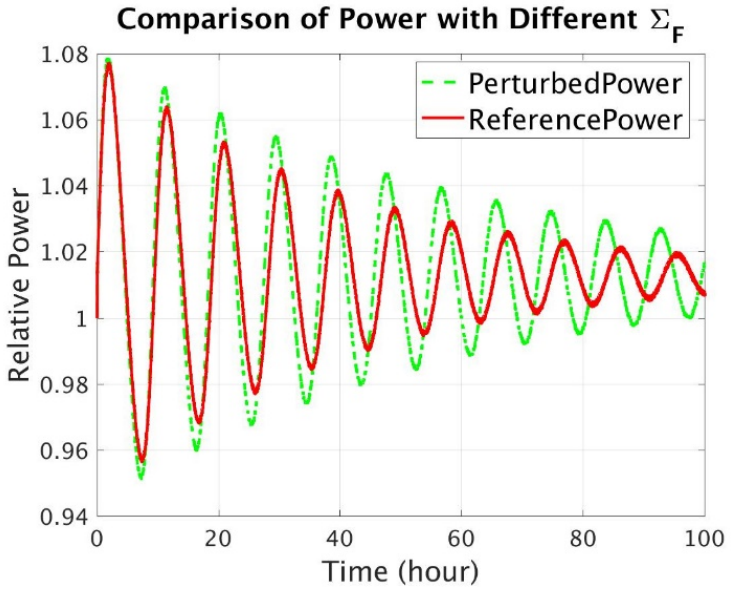

Figure 5.2 Power Sensitivity due to Parameter Σ_F

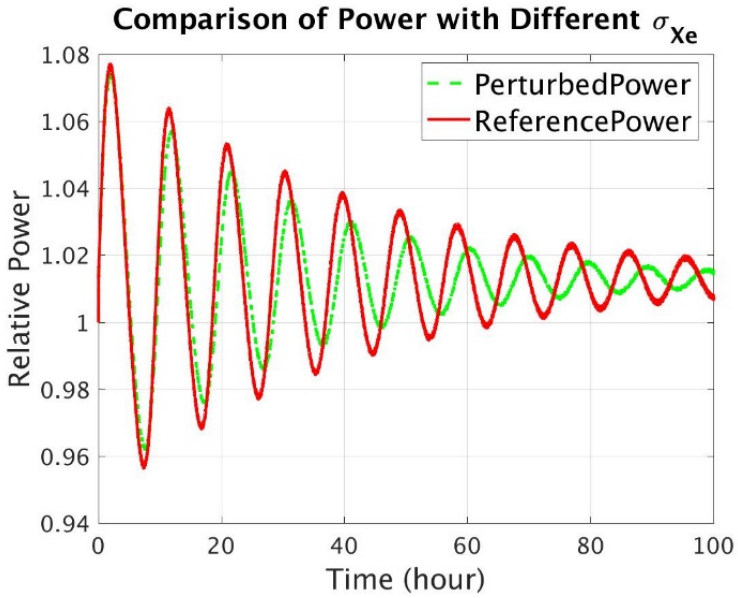

Figure 5.3 Power Sensitivity due to Parameter σ_{Xe}

To self-learn these three parameters, the problem may be cast as an inverse problem, since one is interested in estimating parameters input to a model using the observed model output. A great deal of literature exists on how to solve such problems including basic least-squares minimization, L1 minimization, regularization-based techniques, etc. The choice typically depends on the expected noise in the measurement, the parametrization process as well as the stiffness of the physics equations. A customized inference process is developed here based on an iterative process starting with the simplest approach, e.g., least-squares, and its regularized version, which were found to be inadequate. Interestingly though, going through that process provides proof that it is indeed difficult for attackers to learn a system behavior if they treat the model as a black-box, which certainly supports the premise of model-based defenses. The proposed inference methodology relies on a concerted use of Fourier transform, alternating conditional estimation (ACE), and regularization techniques. We provide a short discussion of ACE here given that it is a critical component of the methodology. Details on Fourier transform and regularization are left to references [115][116].

5.2.1 Alternating Condition Estimation

Solving an inverse problem requires an inexpensive alternative to the forward model, because it is impossible to find an analytical or numerical description of the inverse model operator for almost

all real-world systems. Therefore, one must be able to execute the forward model efficiently to explore the effect of changing the model parameters on the responses of interest. This is true as the number of model parameters increases, and the cost of the forward model becomes prohibitive for repeated executions. If a simplified physics-based approximation exists for the forward model, this is considered to be the best approach and the simplified model is referred to as a low-fidelity model as opposed to the high-fidelity model representing the best available approximation of the real system behavior. A lower fidelity model can be constructed from a high-fidelity model by simplifying the equations and/or the numerical discretization scheme via, for example, the use of coarse versus fine meshes. This approach is referred to as physics-based because it attempts to retain the physics principles that underpin the behavior of the system. If this is not possible, one resorts to the use of parametric methods, also referred to as response surface methods. In this latter approach, one assumes a functional form with some unknown features, e.g., undetermined coefficients, and fits that form using training data generated by the high-fidelity model. The fitting is achieved mathematically using a minimization search that identifies the best coefficients to minimize the discrepancies between the high fidelity forward model and the assumed response surface. An excellent example of this class of methods is the commonly used least-squares-based polynomial fitting approach. With different surfaces, a wide class of methods have been proposed over the years. Examples include the use of radial basis functions, polynomial chaos expansion, orthogonal polynomials, etc. In the statistics community, this type of function approximation is typically referred to as supervised learning. Another class of methods that has gained a lot of prominence in the data mining community is the so-called unsupervised learning methods, which employ nonparametric methods to design approximations of the high-fidelity model. Nonparametric methods preclude the need for parametric surface representation. Instead, the approximation is based on employing the training data directly to make predictions. In this work, we employ ACE, one of the most famous nonparametric methods, developed by Friedman in the 1980s, that has since then been further developed by many researchers. The basic implementation of ACE is as follows. The algorithm is provided with training data sets for the model parameters x_1, x_2, and x_3 and the output response y, limited here to a single response for illustration. ACE calculates transforms for each parameter and a model response as follows [117]:

1. Initiate all data, $\theta(y) = y / \|y\|,\ \phi_i(x_i) = 0,\ \|y\| \equiv [E(y)^2]^{1/2}$

2. $e^2(\phi,\theta) = E[\theta(y) - \Sigma_i \phi_i(x_i)]^2$

3. Iterate until $e^2(\phi,\theta)$ fails to decrease:

4. For $k = 1$ to p, do:

$\phi_k'(x_k) = E[\theta(y) - \Sigma_{i \neq k} \phi_i(x_i) | x_k]$;

Replace $\phi_k(x_k)$ with $\phi_k'(x_k)$

End For Loop;

End Inner Iteration Loop;

$\theta'(y) = E[\Sigma_{i=1}^{p} \phi_i(x_i) | y] / \|E[\Sigma_{i=1}^{p} \phi_i(x_i) | y]\|$

Replace $\theta(y)$ with $\theta'(y)$

End Outer Iteration Loop;

$\theta,\ \phi$ are solutions, mentioned as transforms.

5. End ACE Algorithm.

If one is interested in estimating *y* for another set of parameters not used during the training process, the ACE algorithms relies on interpolating the transformations at the given parameter values to determine the predicted response *y*. One can show that the transformations are generated based on maximizing the mutual information between the linearly combined transformed parameters and the transformed response *y*.

5.2.2 Inference Calculational Procedure

The calculation procedure is described below and depicted in Figure 5.4.

1. Starting with estimates for the k parameters ($k = 3$), generate an estimate for the power profile by solving equations (1) through (4). The dimension of the power profile is denoted by *N*, i.e., the number of components, one component per time step.
2. Generate *M* randomized perturbations of the parameters and calculate the corresponding power profiles.
3. Apply fast Fourier transform (FFT) on the *M* power profiles.
4. Using scatter plots and simple variance measures, identify the dominant Fourier coefficients associated with each parameter, where dominance implies strong sensitivity to the input parameters.
5. Combine all identified Fourier coefficients into a *K* component vector.
6. This reduces the inverse problem to one with *k* input parameters and *K* output responses. The goal is to identify the best transfer function relating inputs and outputs.
7. Apply the ACE (Alternating Conditional Expectation) algorithm to help identify the best input-output transfer functions. For this task, given the smoothness of the coefficient variations with the parameter perturbations, a 3rd order polynomial is employed.
8. For a given power shape, one can update the parameters by first identifying the *K* Fourier coefficients, and inverting the transfer function in step 7 to determine necessary adjustment for the parameters.
9. With the fitted functions for transforms and inputs, a numerical solver is employed to find a new value for input, given the transform values of responses as well as the initial estimation guess.

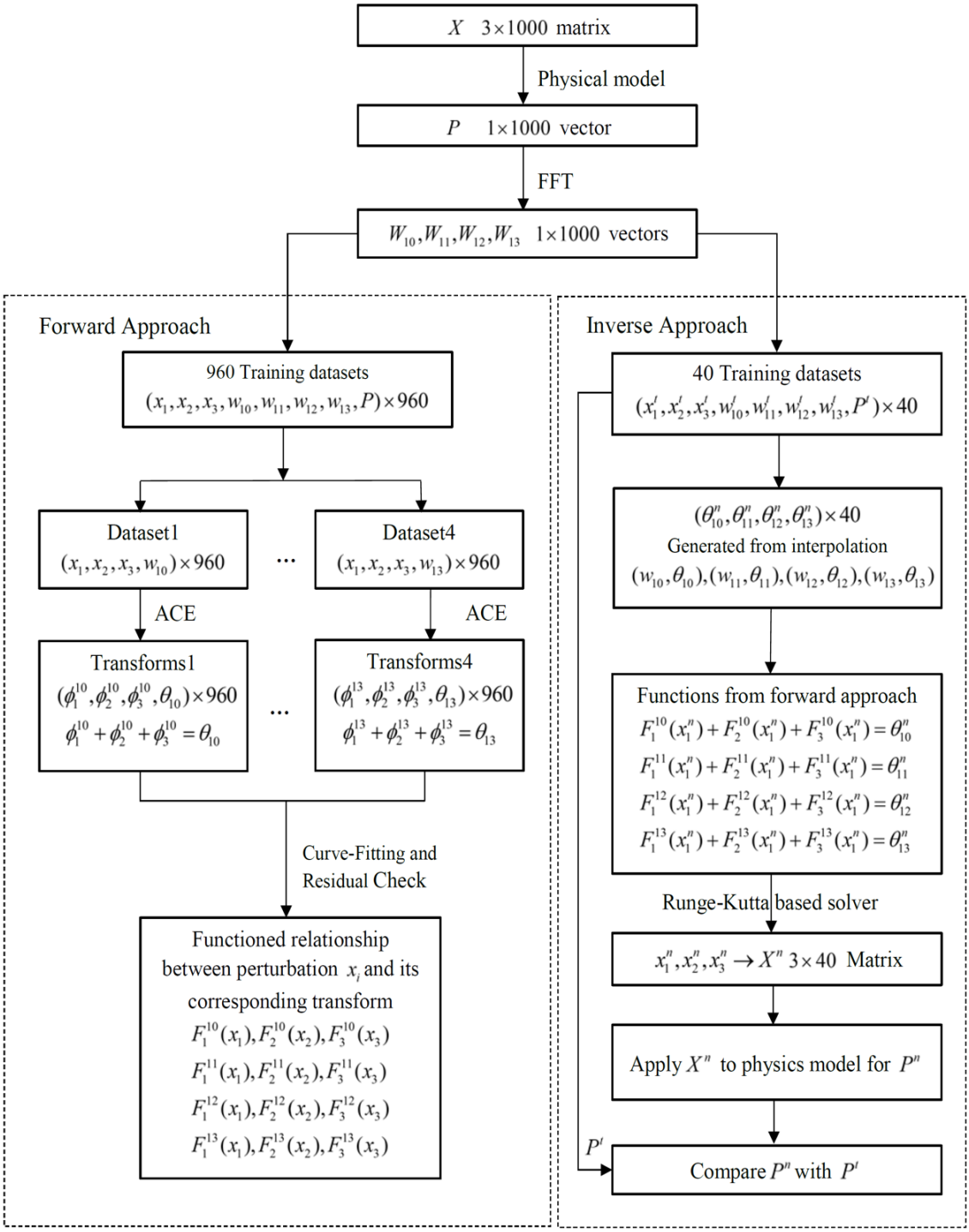

Figure 5.4 Methodology Scheme

Several FFT components are plotted in Figure 5.5, Figure 5.6 and Figure 5.7 to determine which of them can be used to infer the parameters. Our criterion is to pick the FFT components with the highest variations, found to be components 10, 11, 12, and 13. These components are used as responses for the ACE algorithm.

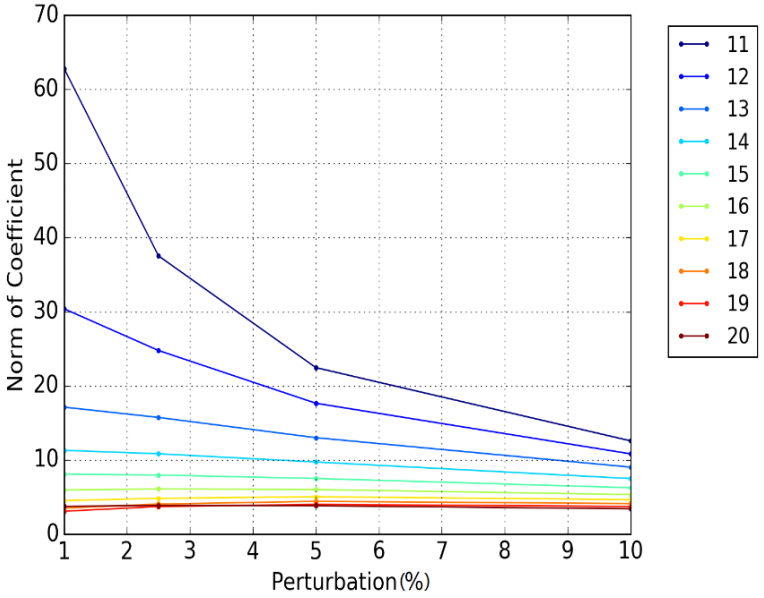

Figure 5.5 Coefficient variations with perturbed α_P

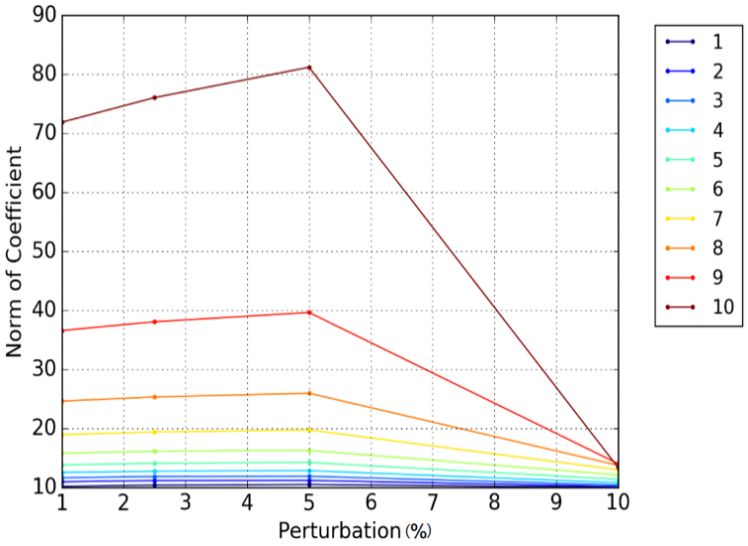

Figure 5.6 Coefficient variations with perturbed Σ_F

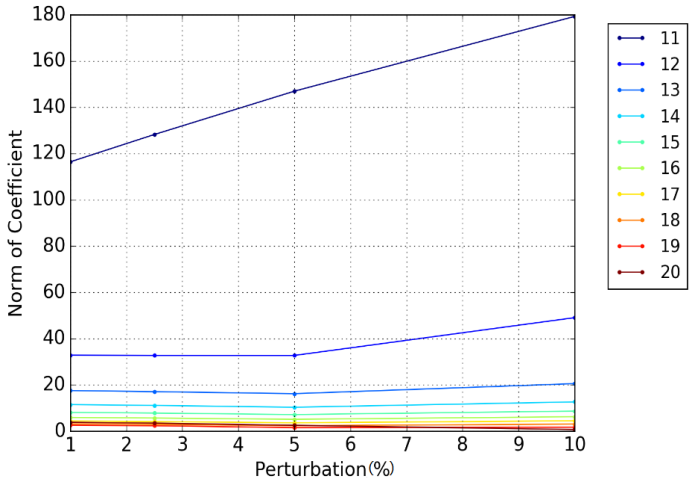

Figure 5.7 Coefficient variations with perturbed σ_{Xe}

The physics model is used to generate 1000 random samples for the responses and model parameters, which are split into two groups, one containing 960 samples for training, and the remaining samples for testing of the ACE model. Each sample consists of three perturbed parameters, and the four FFT components selected for inference. The ACE algorithm employed is based on the python rendition developed by Touran [118]. Figure 5.8 shows representative transformation plots generated from ACE for FFT coefficient #13 as response.

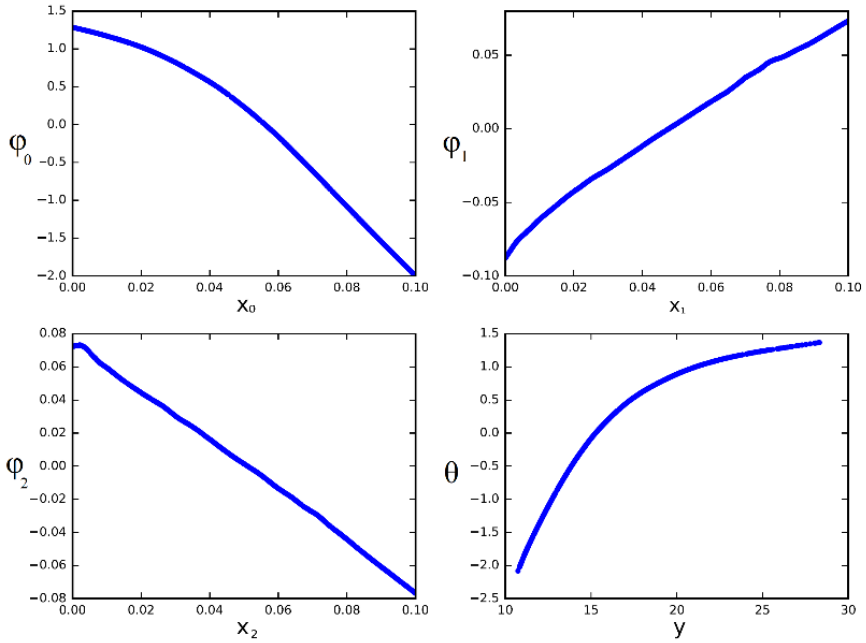

Figure 5.8 Transformation plot of coefficient 13 from ACE

Based on the fitted transform functions, a Runge-Kutta based numerical solver is used to find the optimal values for the perturbed control parameters for all four FFT responses. The estimated perturbation values vs. the real ones of each control parameter are plotted individually in Figure 5.9, Figure 5.10, and Figure 5.11. These results indicate very good inference for the first parameter and less for the second two parameters which are more correlated as would be predicted since they both control the oscillatory behavior. The inferred parameters are used to predict the power using the second group of data used for testing. Figure 5.12 shows the comparison of ACE predicted power and the real power for the 963rd sample, demonstrating the ability to accurately predict behavior, even though some of the inference parameters show some correlation.

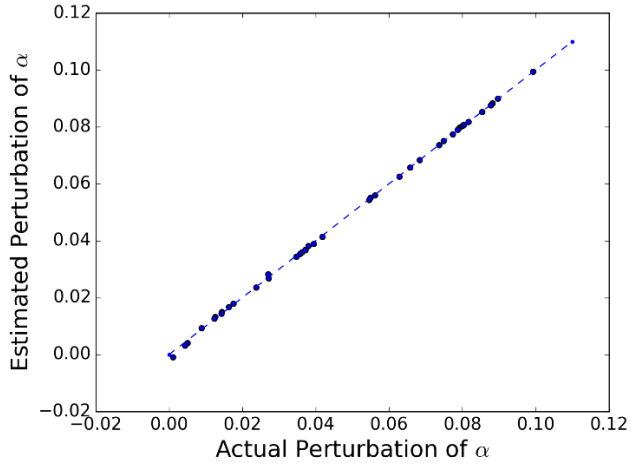

Figure 5.9 Estimated vs. Real Perturbation of α_P

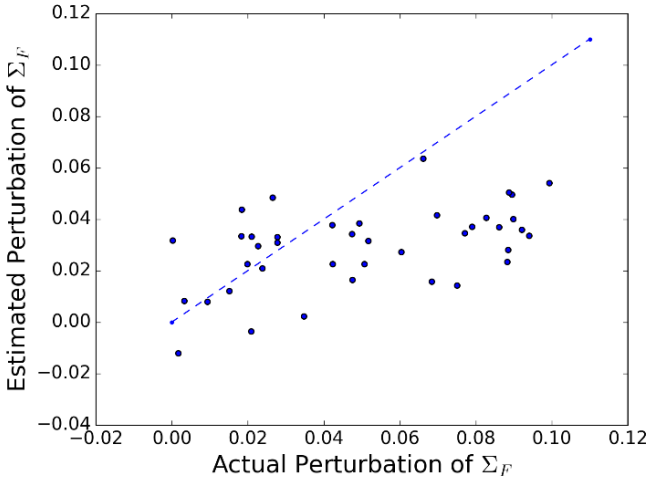

Figure 5.10 Estimated vs. Real Perturbation of Σ_F

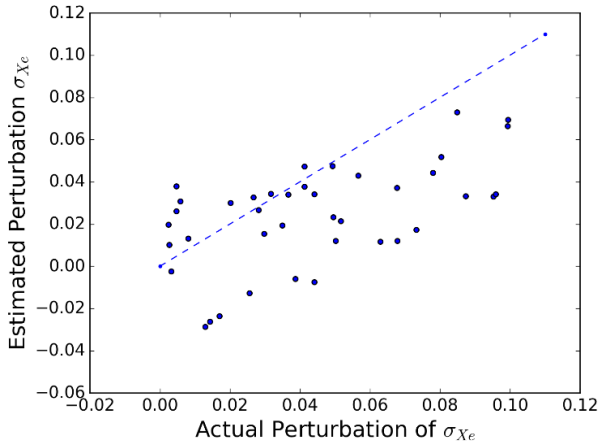

Figure 5.11 Estimated vs. Real Perturbation of σ_{Xe}

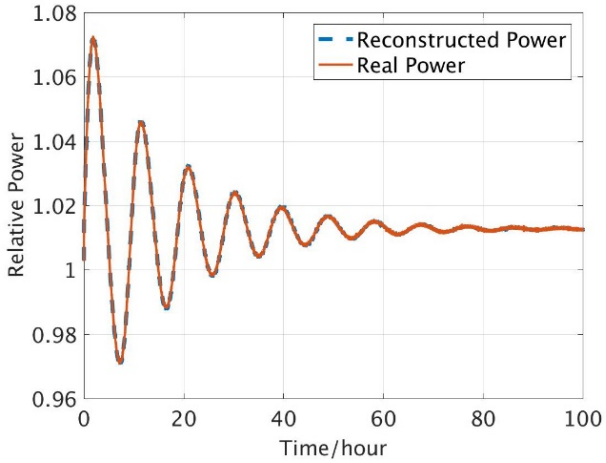

Figure 5.12 Comparison of Reconstructed Power and Real Power

5.3 Data-Driven Adversarial Learning

In order to answer the first question posed in section 1, a pure data-driven approach is employed to learn reactor behavior without any access to the physics model. This is achieved via the use of the MATLAB Deep Learning toolbox [119]. The behavior is learned by using both a single-layered and multiple-layered neural network (NN) approach, shown in Figure 5.13. This is done to investigate whether the addition of more layers could improve the learning process. The multi-layered NN (also referred to as deep NN) is composed of three layers with six neurons in each layer. Like before, the first group of samples is used for training, and the second smaller set for testing. Four different training algorithms and different number of layers are employed to test the ability of data-driven techniques. Specifically, the following algorithms are tested: the resilient

backpropagation algorithm, the one step secant algorithm, the scaled conjugate gradient algorithm, and the conjugate gradient with Powell. The best results are shown in Figure 5.14 Figure 5.15, which compare performance using single, 3, and 5-layered NNs against the model-based results from section IV. In Figure 5.14, the real power is shown in yellow; the neural network reconstructed power is shown in blue; and the model-based power from section IV is shown in red. The subplot titled 'Error' shows the reconstructed error which clearly demonstrates the superior prediction ability of the model-based approach. Figure 5.15 compares the performance of the neural network with different numbers of layers. Results indicate only minor improvement is possible with added layers, none of which however reaches the prediction performance of the model-based approach.

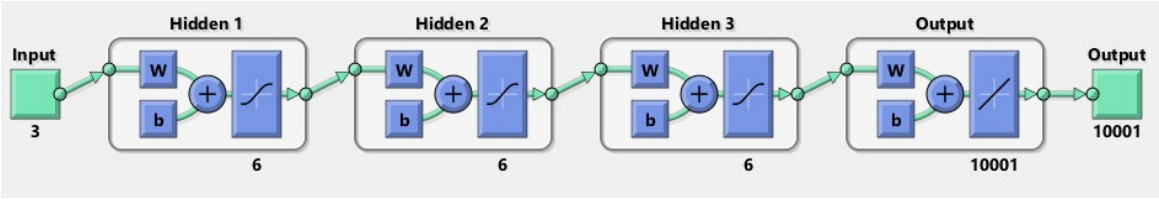

Figure 5.13 Structure of deep neural network

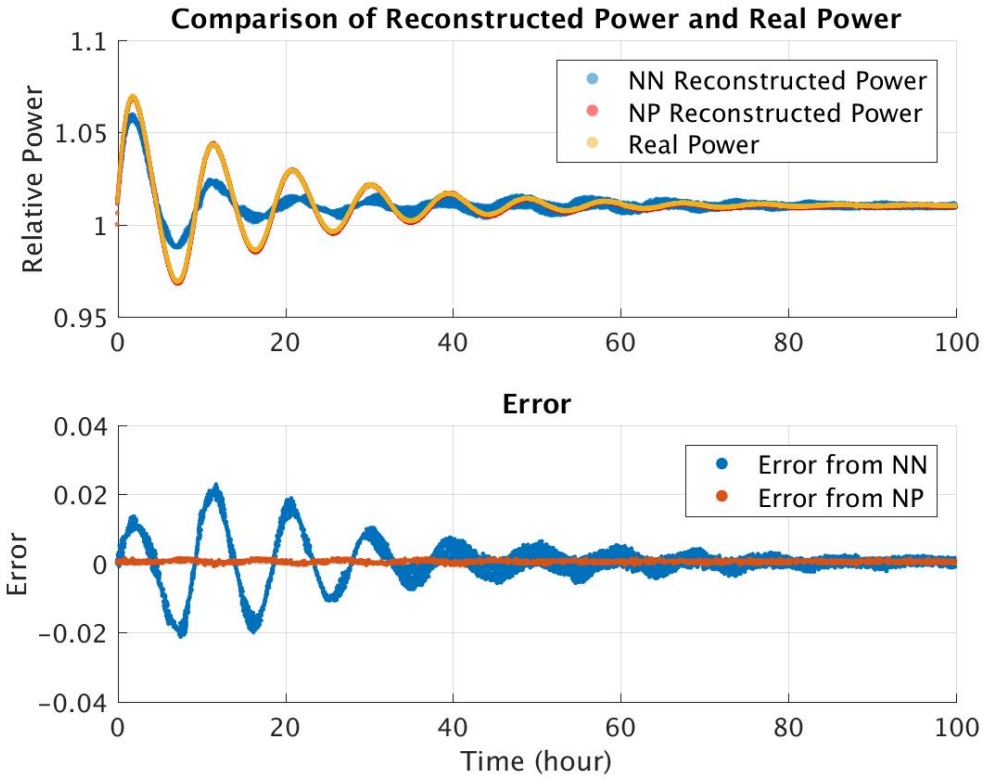

Figure 5.14 Comparison of Power and Errors

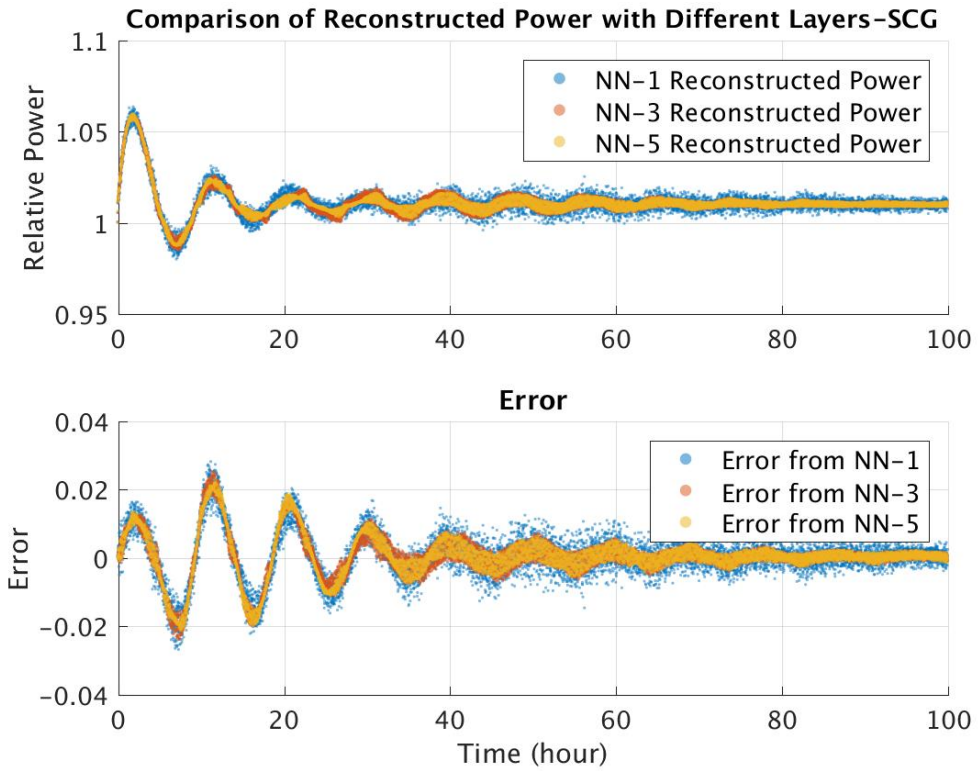

Figure 5.15 Comparison of Power and Errors with Different Layers of Neural Network

5.4 Results Summary

This comparison study demonstrated that it is indeed difficult to develop a predictive model for reactor behavior by relying solely on data-driven techniques, e.g., machine learning. However, with knowledge of the physics, it becomes possible to accurately learn the system behavior. In defense of model-based techniques, we note that it was indeed difficult to employ an off-the-shelf inference capability to predict the true model parameters, and we had to resort to a complicated use of multiple techniques, e.g., FFT, LS, ACE, and regularization. However, considering that the attackers are state-sponsored and can be assumed to have unlimited resources, it is reasonable to assume that attackers will be able to create high fidelity predictive models of the target systems. The results here lead this research to the direction of model-based defense towards the knowledge-based/ stealthy FDI.

6 EXPLORATORY STUDY 1: ALGORITHM FOR STEALTHY FDI ATTACKS [42]

(A version of this chapter has been previously published in Progress in Nuclear Energy, with DOI: 10.1016/j.pnucene.2020.103612.)

The third question stated in the beginning will be answered in this part by a new algorithm that can detect the FDI attacks without triggering the traditional alarms. In the previous section, the signature construction is based on an approximate model to show that relying on a physics model, the attackers can establish a basis for the reactor normal behavior. However the proposed algorithm in this section is based on a novel idea to build signatures, and its implementation is inspired by the dynamic mode decomposition (DMD) algorithm [120]. The idea is that any learning algorithm, whether parametric or non-parametric, supervised or unsupervised, attempts to identify dominant behavior. For example, in reduced order modeling (ROM) relying on the use of singular value decomposition (SVD), the dimensionality of the data is achieved by transforming the data using the most dominant components identified by SVD. The criterion employed to select the number of dominant components is that the reconstructed data are close enough to the original data, with the closeness measured in terms of an error metric, e.g., Euclidean norm, whose magnitude is taken to be of the same order as the acceptable level of error in the process that generated the data. For example, if the data are generated using a physics model where the modeling uncertainties are expected to be in the order of 0.1%, an acceptable criterion for the error metric magnitude would be in the order of 0.1%.

6.1 Mathematical Development

Unlike popular techniques stated in background, the proposed approach employs the higher order components (HOCs) which are typically discarded by standard ROM techniques. The dominant components will be referred to as the low-order components (LOCs). In the statistical and data mining communities, the LOCs are typically referred to as dominant or influential degrees of freedom or principal components, e.g., the first few components in a principal component analysis. In our approach, both the HOCs and LOCs are employed to build signature-based classifiers for normal and malicious behavior. This is essential, because the defender must look for signatures

that are difficult to duplicate by the attacker. The HOCs achieve that purpose because, unlike the LOCs, which can be captured using approximate models, they are expected to be sensitive to all the specific details about the system behavior, which are assumed unknown to the attacker. By way of an example, consider an event that results in an increase in the core flow rate, simulated using both an approximate model (that is available to the attacker) and a high-fidelity model (owned by the defender). If one expands the resulting power distribution variations using a modal analysis, one would expect the first few modes to closely agree as predicted by both models, because essentially both models attempt to capture the dominant reactor behavior. The higher order modes however, representing the HOCs, will be discrepant due to the inherent differences between the two models, which are not all known to the attacker. Interestingly, these differences are not only sensitive to the "proprietary" design details but also to the modeling errors resulting from all modeling assumptions and numerical approximations inherent in the defender's model.

Both the LOCs and HOCs can be readily captured using ROM techniques. To put this in perspective, Figure 1 shows the components of three typical sensors variations as projected onto the components identified by a typical ROM technique. The x-axis is an index for the respective components, and y may be thought of as a linear transformation of the variations over both space and time. The small components in the yellow box are discarded by ROM as very small (i.e., assumed below error criterion). The most dominant components in the blue box, i.e., LOCs, are the components expected to be known to the attacker, as they can be captured by approximate models. This leaves the intermediate components to serve as defense classifiers. Also, it is important to note that the space of HOCs for most reactor models is expected to be much larger than the space of LOCs. This has been repeatedly shown by earlier research, where the LOCs are several orders of magnitude smaller in number than the nominal dimensionality of the data [121].

In support of searching for signature-based classifiers, the application of ROM is essential to reduce the number of LOCs and HOCs employed to build classifiers. This follows because the majority of classification techniques suffer from the curse of dimensionality [122], that is an exponential increase in computational demands with the dimensions of the training data used to calculate the classifiers. This means that the defender must limit the number of HOCs terms employed in the construction of signatures. While a limitation at a first glance, this provides great

strength for the proposed active OT defense, because now the attacker has to guess which of the HOCs have been used by the defender to design signatures. This is nearly an impossible task given the huge size of the HOCs space as discussed earlier.

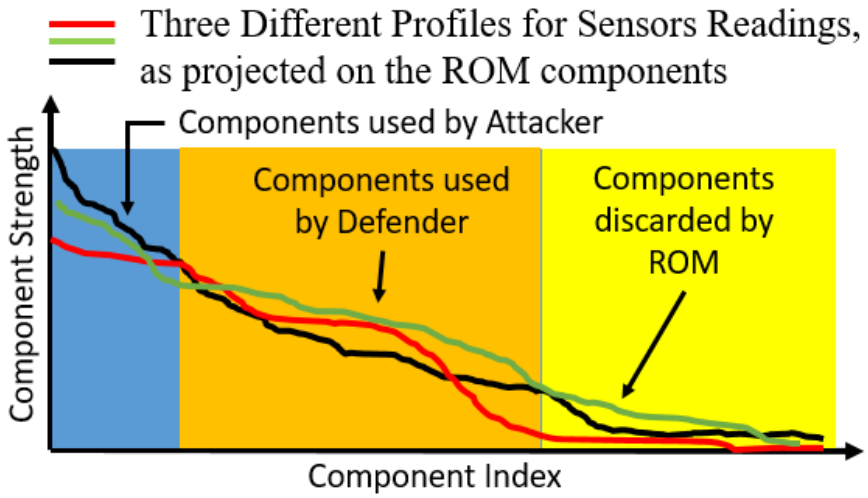

Figure 6.1 ROM Components Availability Illustration

To capture the LOCs and HOCs components, we rely on a new RWD algorithm which is inspired by the DMD algorithm [123] discussed below. The synthesis of signatures using the identified LOCs and HOCs is achieved using support vector machines (SVM). Each of these enabling algorithms are discussed next. The following section discussed the overall implementation of the RWD algorithm and associated signature-based classifier construction.

6.1.1 Dynamic Mode Decomposition (DMD)

Dynamic mode decomposition is a popular method to construct an analytic emulator (i.e., surrogate model) to be used in lieu of a dynamical system model [123]. For physics models exhibiting no feedback, one can show that the emulator predictions are exact. The implementation is seamlessly achieved by taking a time series generated by the physics model and turning it into a matrix by running a window of fixed size over the data, with the window size equal to the number of columns of the resulting matrix. Every row corresponds to a shift of the window over the time series by a fixed time step, typically taken to be a single time step. If the time signals have n time steps, and the window size is k wide, the resulting matrix, $\mathbf{X}$ would be $k \times (n-k+1)$, where $(n-k+1)$

is taken as the collected dynamic modes along with the temporal data. If we take the first *(n-k)* columns to construct a new matrix $\mathbf{X_1}$, and the last *(n-k)* columns to obtain another matrix $\mathbf{X_2}$, which is the time-shifted snapshot matrix of $\mathbf{X_1}$, then the dynamic behavior can be modeled via a constant matrix **A** as given by Eq. (24)

$$\mathbf{X}_1 = \mathbf{A}\mathbf{X}_2 \tag{24}$$

Then apply truncated singular value decomposition on operator **A**, and get three matrices, $\tilde{\mathbf{U}}$, $\tilde{\mathbf{S}}$, and $\tilde{\mathbf{V}}^{\mathbf{T}}$. Then construct the reduced-order operator $\tilde{\mathbf{A}}$ in Eq. (25):

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\mathbf{X}_1\tilde{\mathbf{V}}^{\mathbf{T}}\tilde{\mathbf{S}}^{-1} \tag{25}$$

The eigen-decomposition of $\tilde{\mathbf{A}}$ in Eq. (26) yields eigenvalue and eigenvectors, which can be investigated to understand the fundamental characteristics of the underlying system, like unstable growth mode, etc.

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda \tag{26}$$

The dynamic mode, Φ, is calculated by Eq. (27)

$$\Phi = \mathbf{X}'\mathbf{V}\mathbf{S}^{-1}\mathbf{W} \tag{27}$$

The basic assumption here is that the matrix **A**, representing the mathematical operator for the time evolution, is constant in time. For most practical problems, e.g., reactor analysis, the physics feedback forces the operator **A** to be a function of the time evolution of the solution. Several strategies to address this have been proposed [124], but are outside the scope of this project. The proposed RWD algorithm constructs a single matrix **X** by placing the window at random points over the temporal scale and over all the snapshots of the solution obtained from repeated

randomized execution. The total number of rows of the matrix are much smaller than the number of time steps and the number of executions. The goal here is not to compare against DMD algorithm, since the objective is not to construct a surrogate model. Instead, the goal to efficiently identify a number of LOCs ad HOCs that can be used for signature-based classification of behavior.

6.1.2 Randomized Window Decomposition (RWD)

The idea of RWD is to randomize the placement of the window over all the snapshots of the response's temporal behavior as opposed to a sequential movement of the window over a single time series as performed by DMD. Different size windows can be employed. The longest window corresponds to the length of the time series, which reduces to conventional SVD decomposition, where the entire time series represents a single snapshot. Very short time windows are not expected to provide useful information about system behavior, given the small number of degrees of freedom available. Thus, some experimentation is required to identify a proper size window. From a defender viewpoint, this experimentation adds another level of obscurity to the design of the defense algorithm.

The input data processed by the RWD algorithms are snapshots of the time series for the given responses, obtained via multiple executions of the software under a wide range of conditions expected during normal and/or abnormal operation. This can be achieved by rerunning the simulation under different scenarios and randomly perturbing relevant initial and boundary conditions within expected ranges of variations. The i^{th} simulation generates a temporal variation for a given response, denoted by y_i, with y_0 representing the reference temporal variation. The window size is denoted by w. Collect snapshots for the response of interest over time and aggregate in a matrix $\mathbf{G}^d$ of size $t \times L$, where t is the number of time steps for the response of interest, and L is the number of model executions, with each execution generating a different temporal response based on perturbed initial and boundary conditions, i.e., the i^{th} column of $\mathbf{G}^d$ is given by y_i. Next, standardize the matrix $\mathbf{G}^d$ by subtracting and dividing by y_0.

Given the attacker's familiarity with the system, we assume that the attacker can approximate the matrix $\mathbf{G}^d$, denoted by $\mathbf{G}^a$, where d and a refer respectively to defender and attacker.

1. Employing a window of size w, randomly place the window over y_0 to generate n_0 random snapshots for the window values, and aggregate in a matrix $\mathbf{D}_0$ ($w \times n_0$). Generalization of this could be achieved by placing the window randomly over all the columns of the matrix $\mathbf{G}^d$.
2. Apply SVD on $\mathbf{D}$ to determine the rank r_L (based on a defined tolerance), LOCs, and HOCs, captured in two matrices $\mathbf{U}_L$ ($w \times r_L$), and $\mathbf{U}_H$ ($w \times r_H$). The size of the HOCs matrix r_H is arbitrary as those components are typically discarded by SVD as non-influential.
3. Employing a window of the same size w, generate a matrix $\mathbf{D}_i$ of size ($w \times n$) corresponding to the i^{th} column of the matrix $\mathbf{G}^{\text{d}}$ run, where n can in general be different from n_0.
4. Calculate the projection of each of the n windows from the $\mathbf{D}_i$ matrix along the r_L LOCs and the r_H HOCs determined using $\mathbf{D}_0$. This generates two matrices of sizes $r_H \times n$ and $r_L \times n$, denoted respectively by, $\mathbf{\Pi}_H^d$ and $\mathbf{\Pi}_L^d$.
5. Repeat the above steps using the attacker matrix $\mathbf{G}^a$ to generate matrices $\mathbf{\Pi}_H^a$ and $\mathbf{\Pi}_L^a$.
6. Using a binary SVM classifier, identify signatures to attack scenarios.

6.2 Application Demonstration – Subtle FDI Detection

This section applies the methodology described above to a number of representative scenarios during operation. The goal is to distinguish between normal behavior and FDI attacks and the system components with different degradation levels. The system analyzed is two representative PWR models and the RELAP5 simulator is used for estimating system behavior during both scenarios.

Generally speaking, an FDI attack could be introduced to both steady state and transient behavior, with transient behavior being the more likely approach to ensure the FDI signals could be masked as normal operational maneuvering. Hence for this study, we focus on transient behavior which is expected to be normal by the operator. For nuclear reactors, this transient behavior could result from normal power maneuvering to meet the electricity grid demand. Load-following operation is common for the nuclear industry. For example, both in the US and abroad, e.g., France and

Germany, load-follow operational strategies have been adopted to increase the penetration of nuclear power to the overall energy demand [125][126]. A technically-savvy attacker will take advantage of this power maneuver to first learn system behavior and also to hide their FDI signals within the range of variations that is considered acceptable by the operators.

6.2.1 Model Description

Regarding the specific reactor model used for demonstration, a RELAP5 model for a representative PWR reactor is used. It consists of a primary loop and a secondary loop producing a 50 MW as its peak power. The simulation time is set to 200 seconds when the system reaches a steady state. All physics responses are output by RELAP5 every second. The nodalization of this model is shown in Figure 6.2 [127]. Each number represents a given component of the system, where a component may designate an actual physical component, e.g., steam generator, pipe, etc., or a section thereof as dictated by the numerical scheme. To simulate possible response variations, either due to modeling or operational uncertainty, ten parameters associated with different components are selected for perturbations as shown in Table 7.

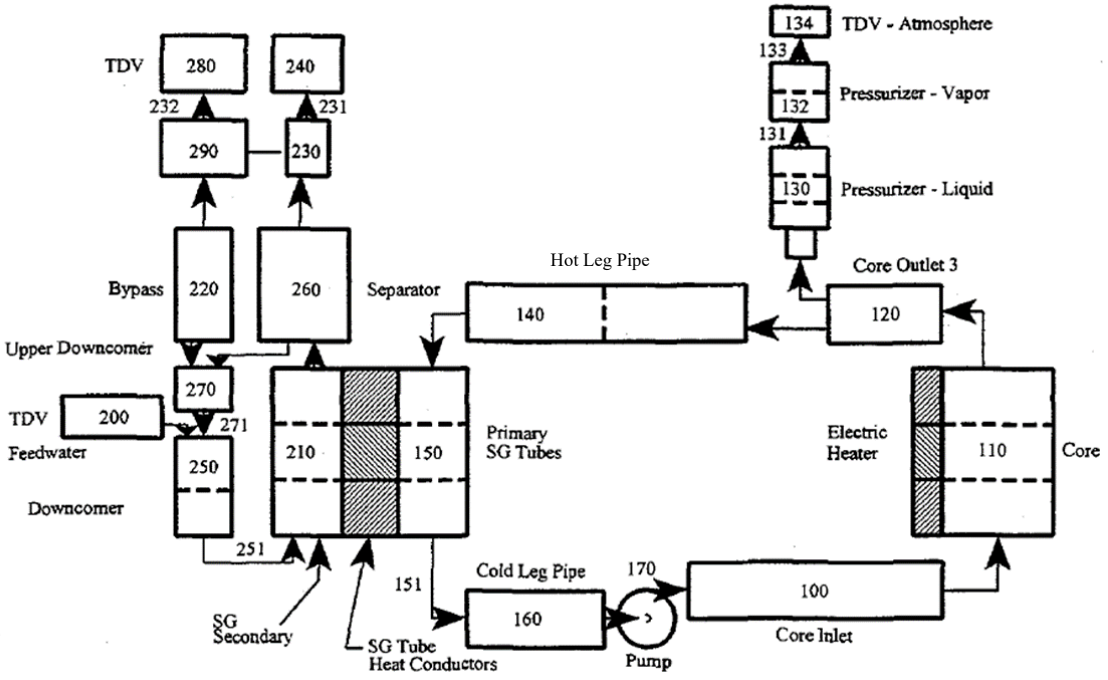

Figure 6.2 Nodalization of RELAP5 Model (source, Ref [127])

Table 7 Perturbed parameter and range

Parameter	Feedback Fuel	Loss Factor F of 210	Loss Factor r of 210	Loss Factor F of 251	Loss Factor r of 251
Symbol	a_f	f_loss_210	r_loss_210	f_loss_251	r_loss_251
Range	(-0.2,0.2)	(-0.1,0.1)	(-0.1,0.1)	(-0.1,0.1)	(-0.1,0.1)
Parameter	Feedback coolant	T_feed water	T_coolant	Power level	T_fuel
Symbol	a_c	T_inlet	T_c	power	T_f
Range	(-0.2,0.2)	(-0.05,0.05)	(-0.1,0.1)	(-0.1,0.1)	(-0.1,0.1)

In general, the attacker is expected to change the time evolution at multiple points during time to achieve their goal of manipulating system state. However, for the sake of developing insight and assessing the efficacy of the proposed algorithm, we assume the attacker changes the trend at a single time window only, which represents the most challenging scenario for the OT defense. To simulate the attack, it is assumed that the attacker has access to a physics model that can approximate the behavior of the system to a reasonable accuracy. As mentioned earlier, this means both the attacker and the defender can approximate the same LOCs. To reproduce this scenario here, the RELAP5 is used as a basis for generating the time evolution of the various responses as would be done by the defender, collected in the matrix $\mathbf{G}^d$. To simulate a triangle attack that captures the LOCs, the time evolution within selected windows, randomly placed over the time horizon for the simulation, is converted to simple linear variations as shown in Figure 3. The right graph shows a representative time evolution for a given response. It is hypothesized that the attack is inserted between the two vertical dashed red lines, where the trend is changed to be linear, which preserves the dominant behavior. Thus, each of the columns of the $\mathbf{G}^a$ matrix is assumed to contain a single attack placed randomly throughout the time horizon for the simulation.

As mentioned above, state-sponsored attackers can duplicate the dominant system behaviors by duplicating LOCs. In this case study, part of the LOCs produced by defenders and attackers can be found in Figure 6.3. The high correlation between the LOCs from defenders and attackers indicates that attackers are able to capture the LOCs almost identically to the defenders'. However, the attacker cannot reproduce HOCs at the same accuracy level as LOCs, which is demonstrated in Figure 6.4.

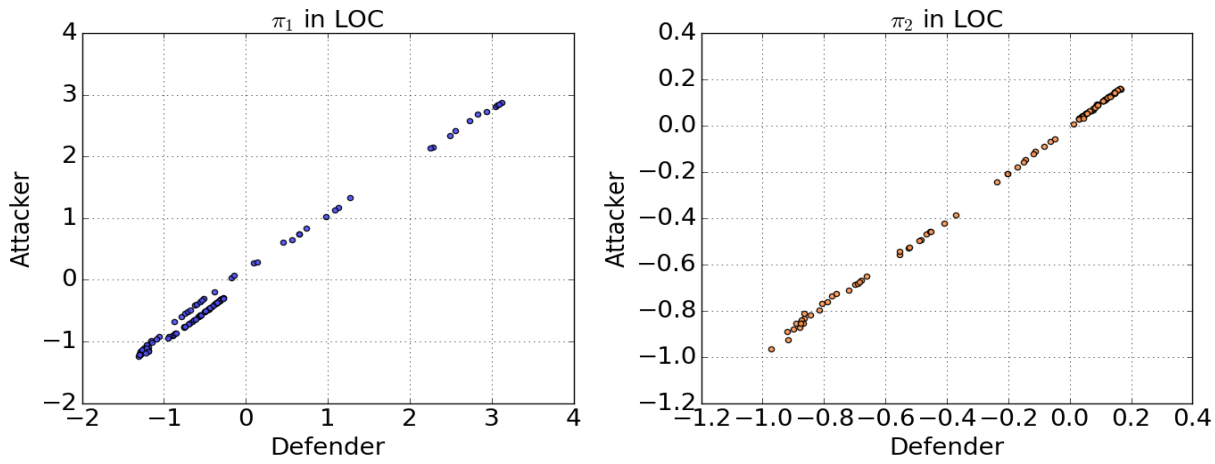

Figure 6.3 LOCs Produce by Defender and Attacker

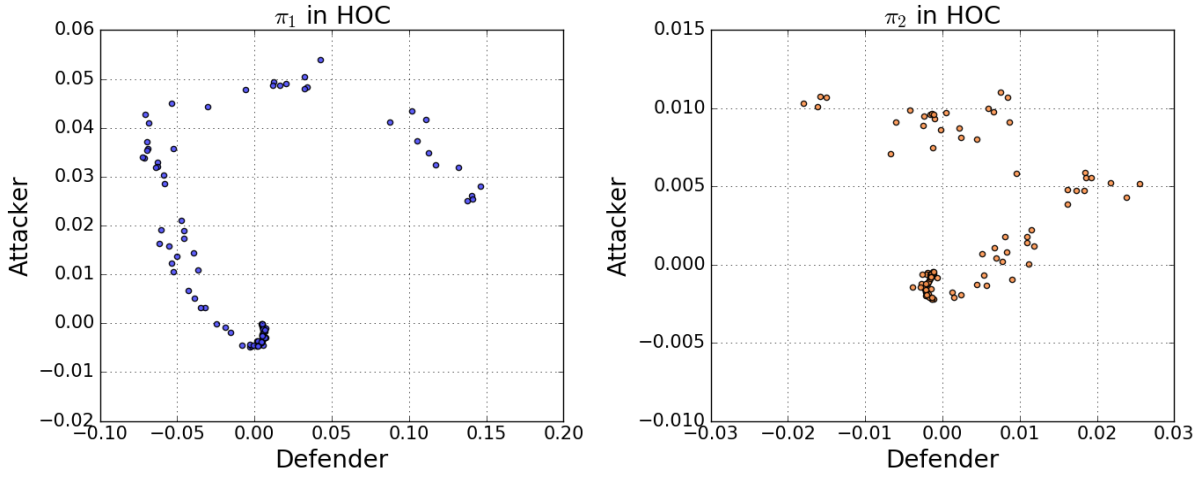

Figure 6.4 HOCs Produced by Defender and Attacker

Next, regarding the choice of the LOCs and HOCs, one can include all LOCs components and an arbitrary number of HOCs to develop the SVM classifier. For this initial study, we focus on employing a single component from each set to help develop insight into the mechanics of the proposed OT defense. Thus, it is assumed that r_L=1 and r_H=1, representing a single LOC and a single HOC component. In this case, the matrices $\mathbf{\Pi}_H^d$ and $\mathbf{\Pi}_H^d$ reduce to two vectors, of length n, where n is the number of time windows placed over the time horizon of the simulation. The overall calculational process is shown in Figure 6.5.

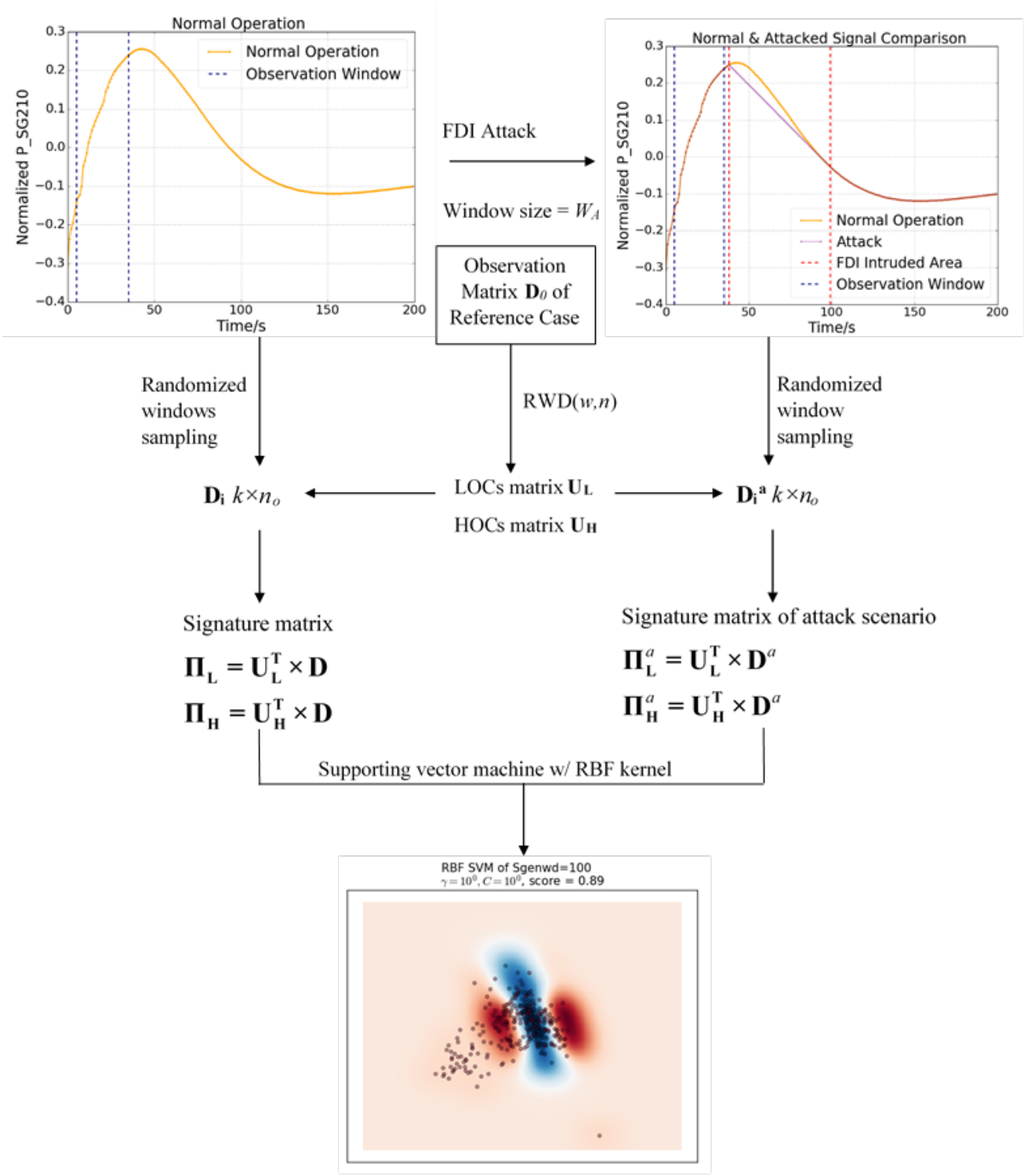

Figure 6.5 Methodology Scheme in FDI Detection Case Study

6.2.2 Numerical Results

In this study, RWD is applied to distinguish between a normal operational scenario (denoted by "True" in the figures' legends) and one infected by an FDI attack (denoted by "Attack"). The calculational scheme in Figure 6.5 generates two sets of features, denoted by π, for either LOCs or HOCs. As one may intuitively think that higher order HOCs will be deployed if the attack has more comprehensive physics model, the components with different orders in HOCs are discussed in two tests. Here the RWD algorithm is applied to each response separately, i.e., both the LOCs and HOCs do not take into account the correlations across difference responses. This will be explored in future work.

In the first numerical experiment, the π_1 HOC is selected for classifier training. Figure 6.6 shows a scatterplot of the selected HOC and LOC for the normal behavior and the FDI-manipulated behavior for the steam generation rate ("Sgen") using a time window of 70 seconds ("wd = 70"). The blue dots ("True") mark the normal behavior, and the red crosses ("Attack") mark the FDI-manipulated behavior. The left subplot shows the evolution of the LOCs and HOCs over the time of the simulation. The right subplot condenses the time-evolution of the LOCs and HOCs using a simple Euclidean norm.

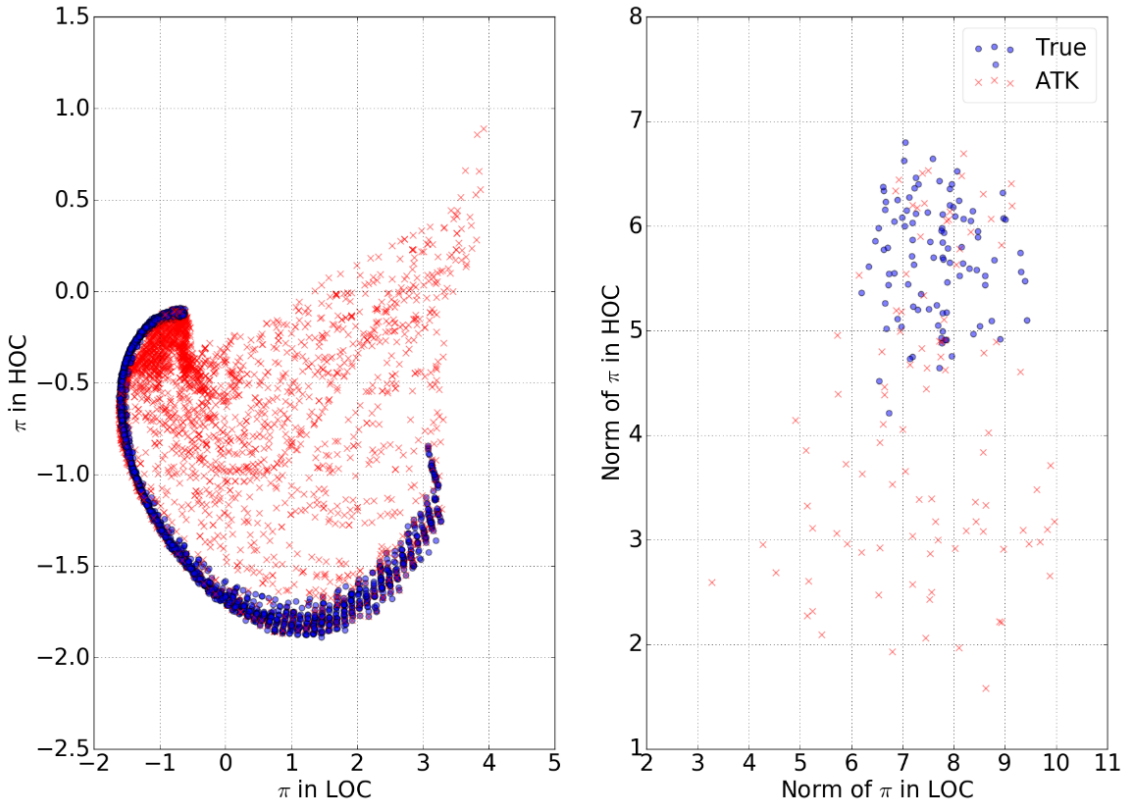

Figure 6.6 Feature of Steam Generation with Observation Window size = 70 (seconds)

These basic results demonstrate the potential of differentiating between normal and FDI scenarios, made possible via the use of HOCs in tandem with LOCs. One can envision many ways to apply the SVM classifiers on these training data. For illustration, we apply SVM on the Euclidean-condensed HOCs and LOCs as plotted in the right subplot of Figure 6.6. The classification results are shown in Figure 6.7 and Figure 6.8 for the steam generation rate when the observation window size is 70 seconds, with the blue area representing normal operation and the red area representing the FDI attack.

The two figures show results for different values of the SVM's parameters γ and C. Weak sensitivity to these parameters is noted. In Figure 6.7Figure 6.7, the values for these two parameters, noted on the graph, result in a 79% classification accuracy, meaning that one can detect the attack in 79% of the cases analyzed. In Figure 6.8, this accuracy of the classification changes slightly to 78%. This weak sensitivity is due to the fact that the HOCs and LOCs have been condensed using Euclidean norm prior to the application of SVM classifier. In general, one would

expect different behavior for the classifier depending on the type and the manner in which the data have been preconditioned. An optimization of the classifier results is certainly needed, however these initial results are intended to show that HOCs provide a unique capability to identify FDI attacks when the LOCs can be accurately captured/learned by the attacker.

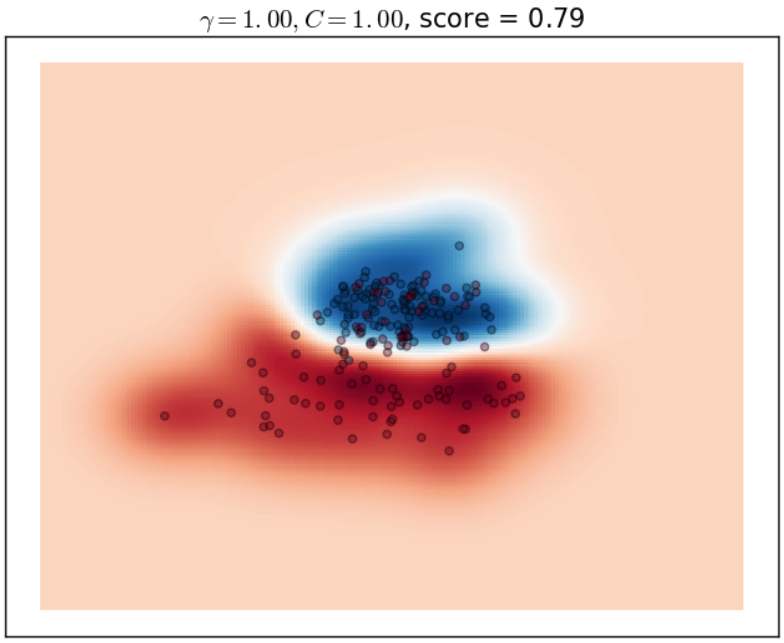

Figure 6.7 Classification Results for the Norm of the Feature of Steam Generation Amount

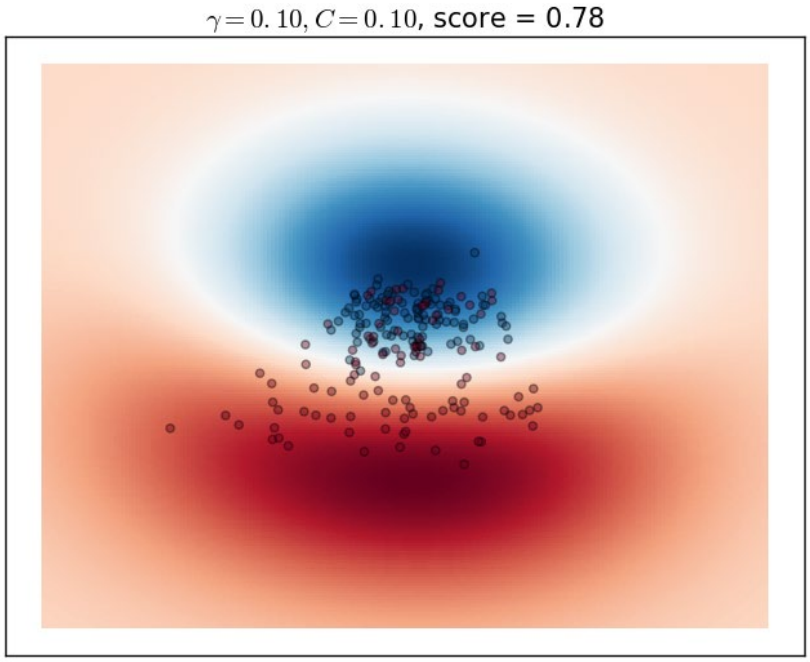

Figure 6.8 Classification Results for the Feature Vector of Steam Generation Amount

Next, Figure 6.9 shows the change in the classification accuracy as the size of the attack window is changed. As one would intuitively think, the classification accuracy will generally improve as the attack window is increased, e.g. pressure and temperature at the secondary side of the steam generator; however, with some noted exceptions, e.g., average core temperature.

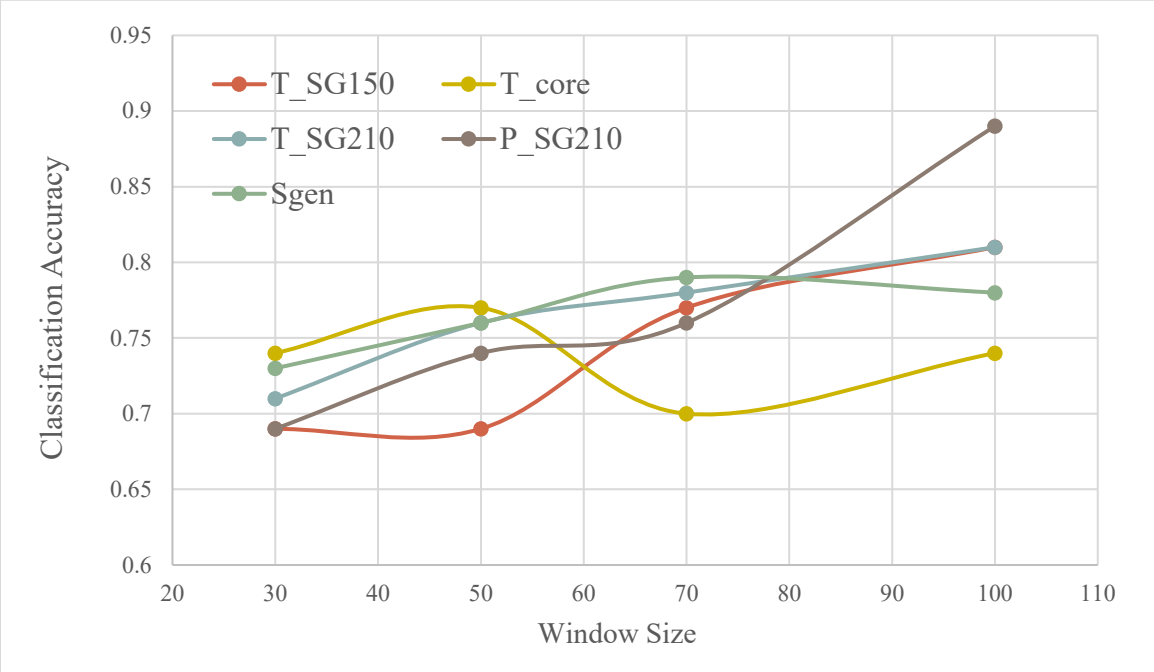

Figure 6.9 Relationship between classification accuracy and window size

Next, the robustness of the HOCs is assessed with respect to process noise, expected to be inherent in all process parameters. The idea here is to assess whether the classification ability based on the use of HOCs will degrade under the presence of noise, expected to be inherent in all the measurements. Previous results employed the RELAP5 simulation results directly as a basis for the training of the classifier. The next set of results repeat the training of the SVM classifier, but now with all data, including both generated by the defender and the attacker, contaminated by white Gaussian noise. Figure 6.10 compares the time evolution of the normal behavior vs. the FDI-manipulated behavior but now with the noise added. Figure 6.11 compares the HOCs and LOCs as done before in Figure 6.6, but now with the noise insertion. Figure 6.12 shows minor changes to the classification accuracy.

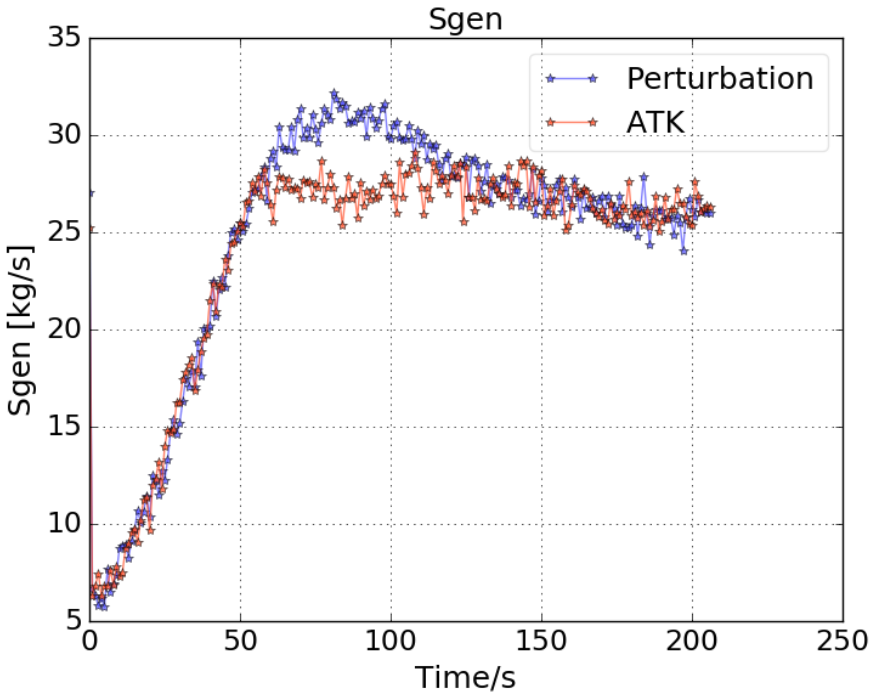

Figure 6.10 Response Comparison with White Gaussian Noise

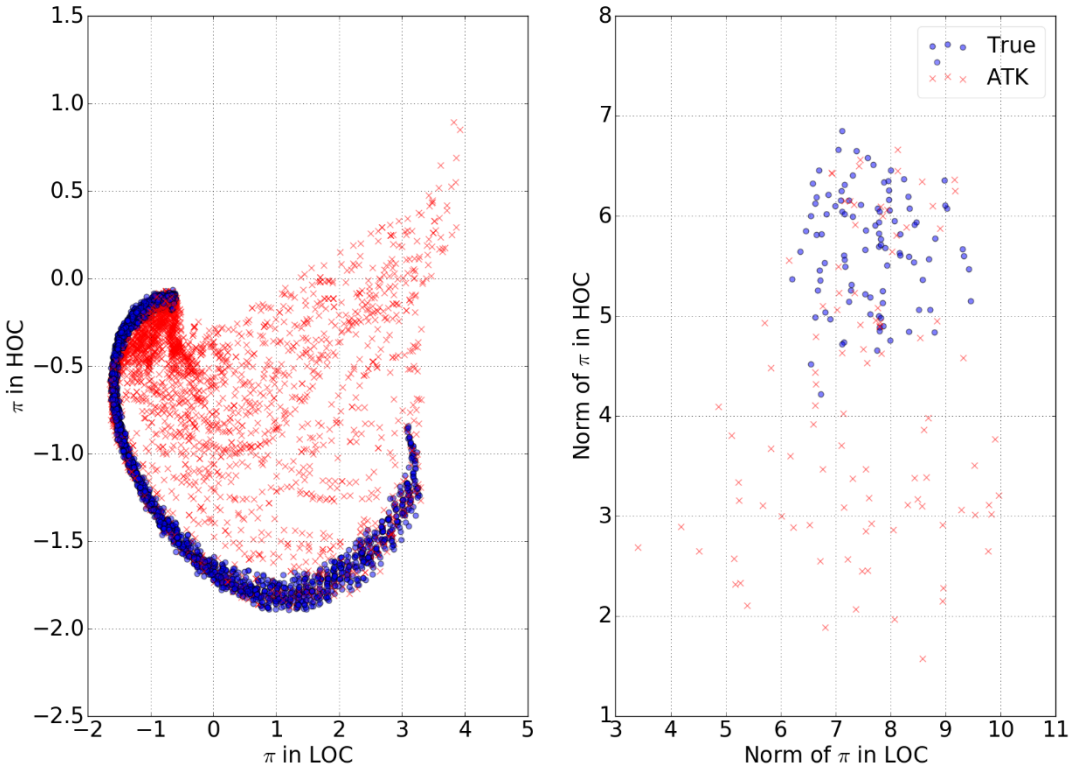

Figure 6.11 Feature of Steam Generation with White Gaussian Noise

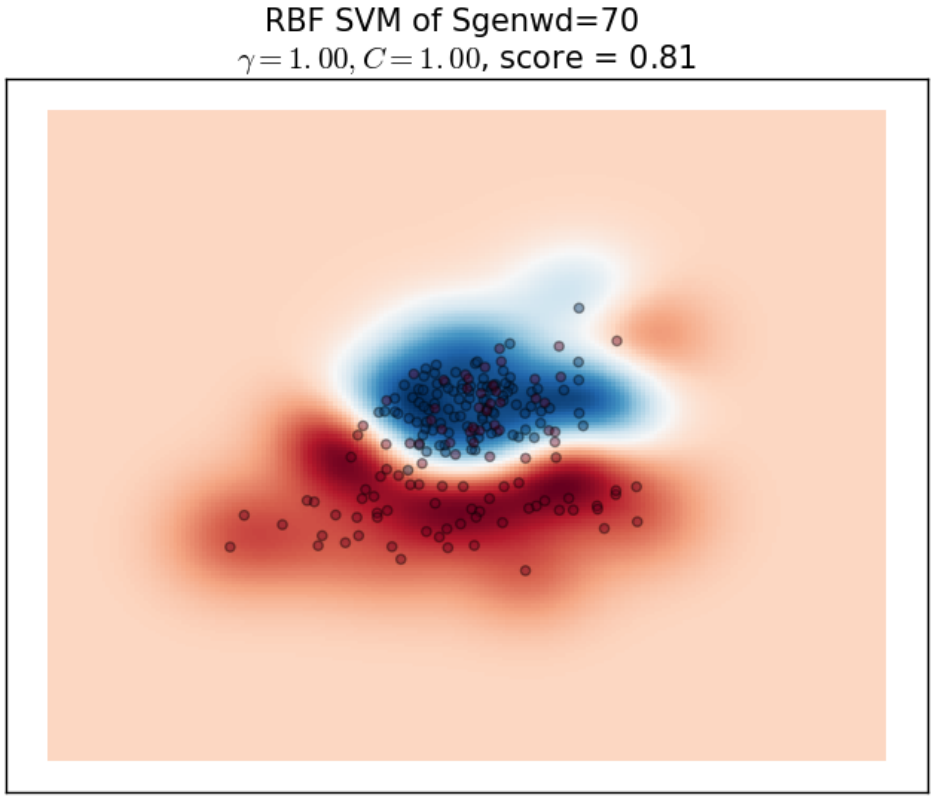

Figure 6.12 Classification Accuracy with White Gaussian Noise

As mentioned earlier, the space of HOCs is expected to be much bigger than that of the LOCs, which provides an additional obscurity defense for the design of the OT defense. The idea is that the defender has a large palette of HOC components to choose from. To demonstrate this, the previous results are repeated but with a different HOC component. Specifically, the fourth component in the HOC set is used for classification, (i.e., th π_4 HOC). Figure 6.13 shows in a similar manner to Figure 6.6 and Figure 6.11 the relationship between the HOCs and LOCs for the normal and FDI scenarios.

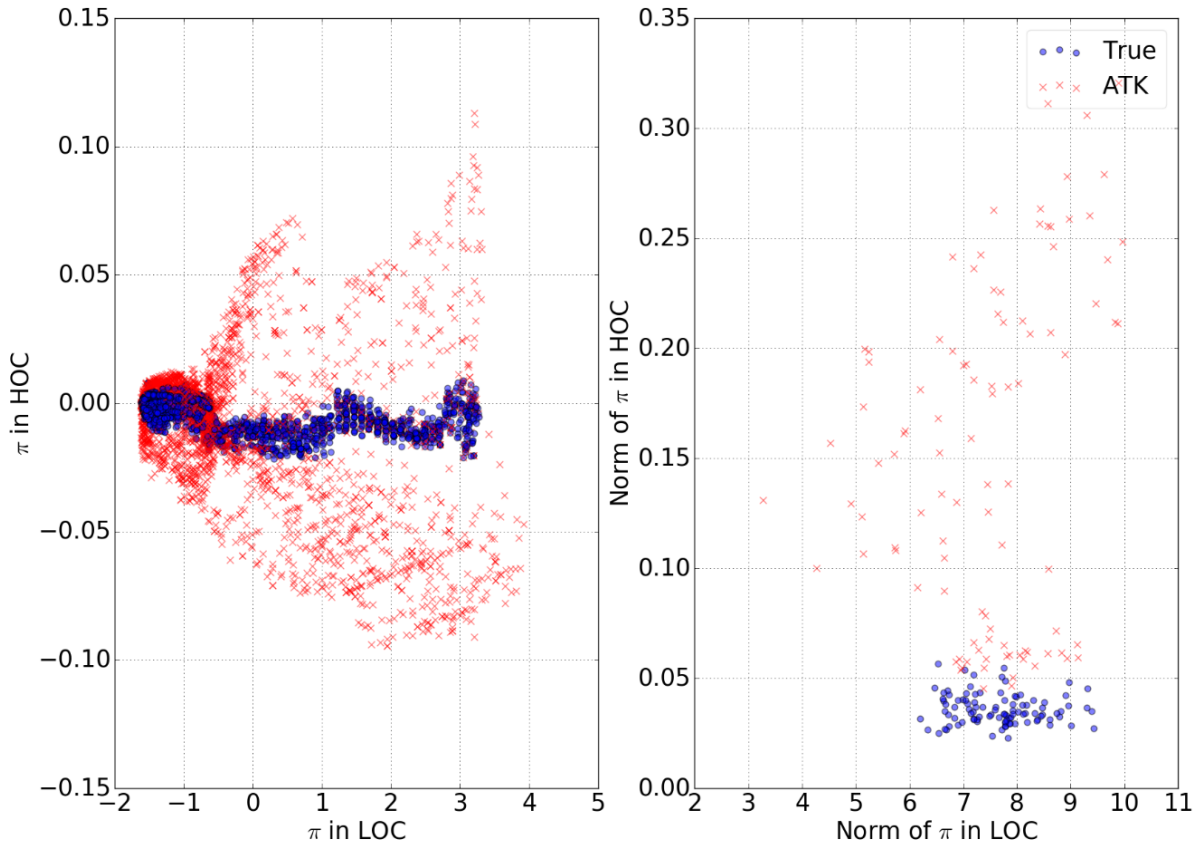

Figure 6.13 Feature of Steam Generation Amount with Observation Window size = 70 (seconds)

The corresponding classification results are shown in Figure 6.14 for the same response, i.e., the steam generation amount with a window size of 70 seconds, with the blue area representing normal operation and the red area for the FDI attack. Figure 6.15 shows the change in the classification accuracy as the size of the attack window is changed in a similar manner to the results shown in Figure 6.9. For these scenarios, the same values for the SVM parameters γ and C are employed, with notable differences in the classification accuracy. Results indicate notable improvement in the classification accuracy as one employs π_4 instead of π_1 for the HOC set. The classification accuracy jumps to 95%. This result implies that the classification accuracy is intimately tied to the way the HOCs are employed to train the classifier. In general, one could use a functional form combining the HOCs components to maximize the classification accuracy.

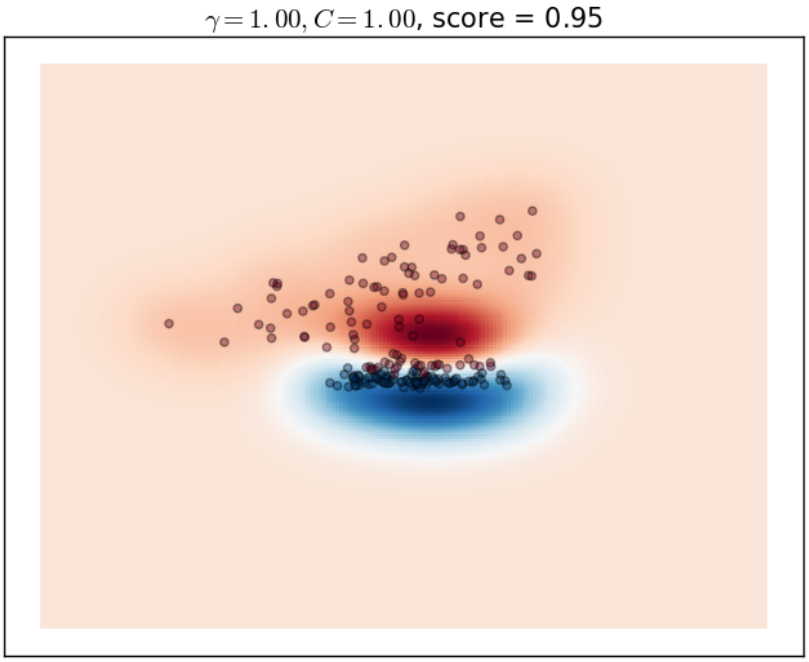

Figure 6.14 Classification Results for the Norm of the Feature of Steam Generation Amount

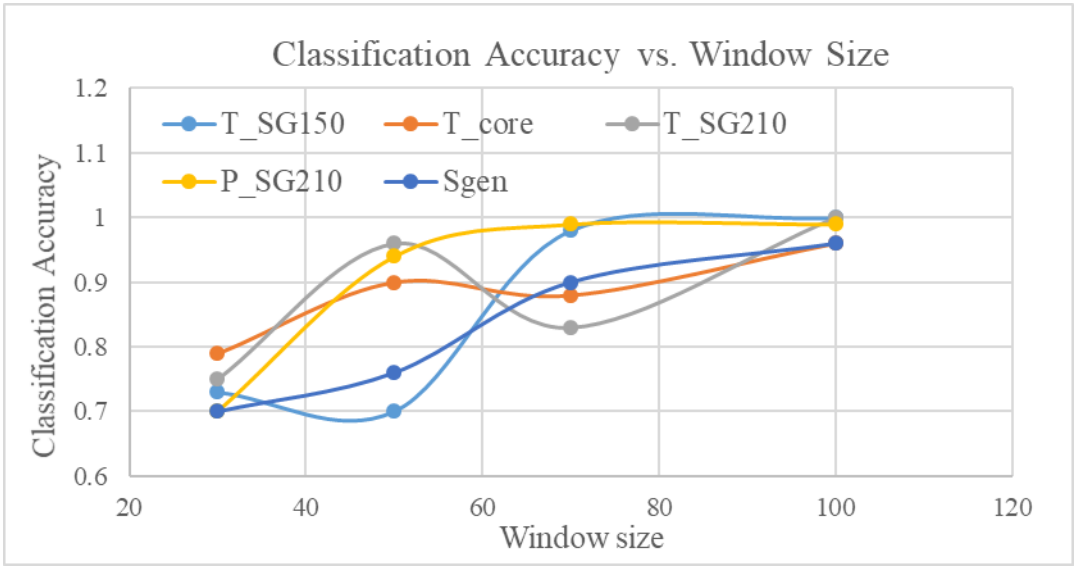

Figure 6.15 Relationship between classification accuracy and window size (with higher order HOC)

6.3 Application Demonstration – Pump Degradation Detection

Besides the detection of FDI attacks during normal operation like power maneuver, the defender needs to accomplish the detection of equipment malfunction. Here, another system is adopted to demonstrate the RWD algorithm in the detection of pump degradation case study, as an envision of the offline analysis.

6.3.1 Model Description

In this case study, a comprehensive RELAP5 simulated PWR with two primary loops is employed to simulate different reactor states. Heat structures are used to represent heat transfer from fuel rods, U-tubes in SG, pressure vessel wall, vessel downcomer wall, core shroud, and internals in the upper head and lower and upper plena. The two primary loops have a slight difference that one represents a single loop, and the other is lumped by three primary coolant loops. Both loops share the same boundary and initial conditions except for the coolant flow rate, one triple the other. This model prints output every second from 0 to 3000 seconds. The nodalization of the model is illustrated in Figure 6.16 and Figure 6.17 [128]. The regular operations with different scenarios are simulated by applying perturbations on the input parameters.

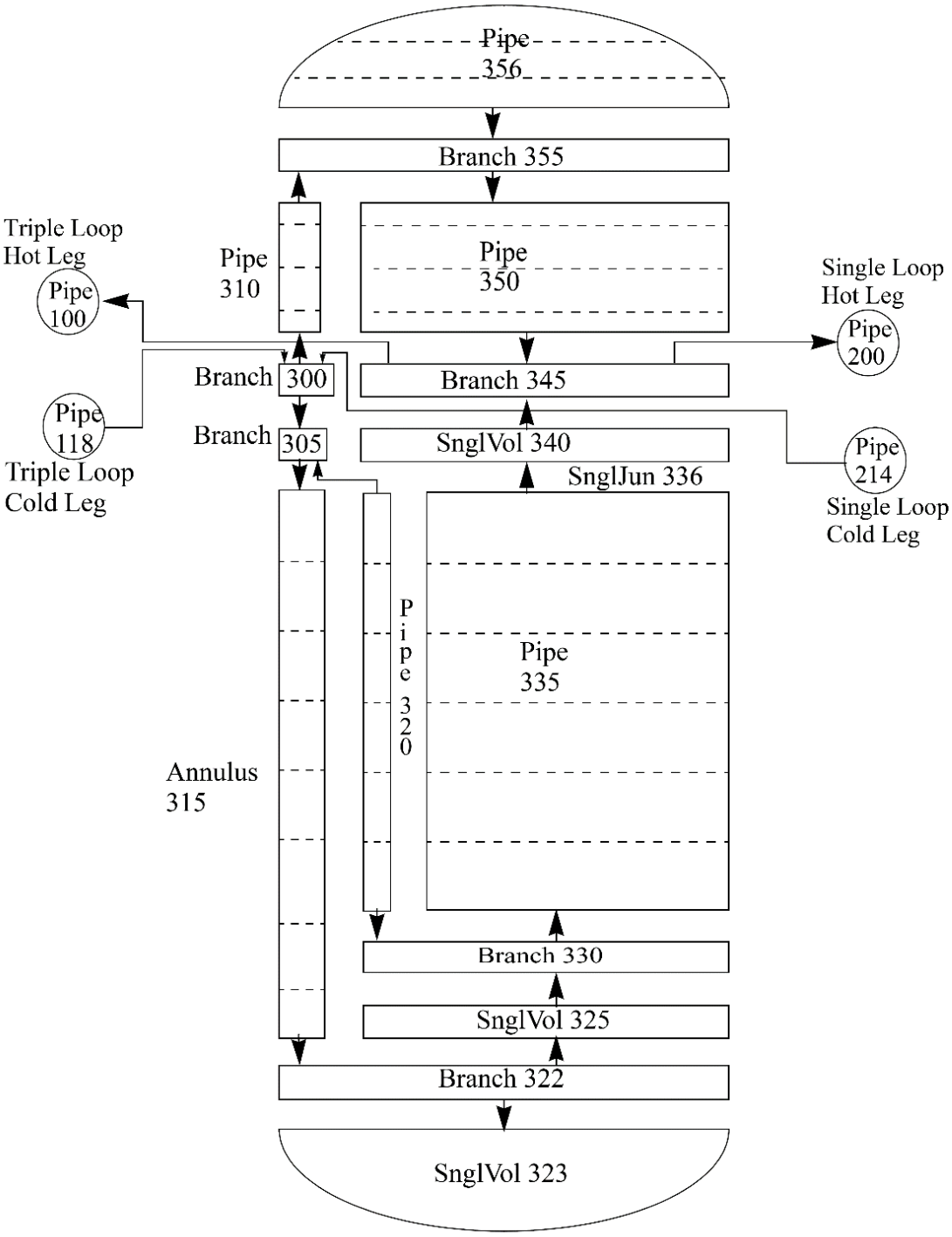

Figure 6.16 RELAP5 Nodalization for PWR: Vessel Model (in Ref. [128])

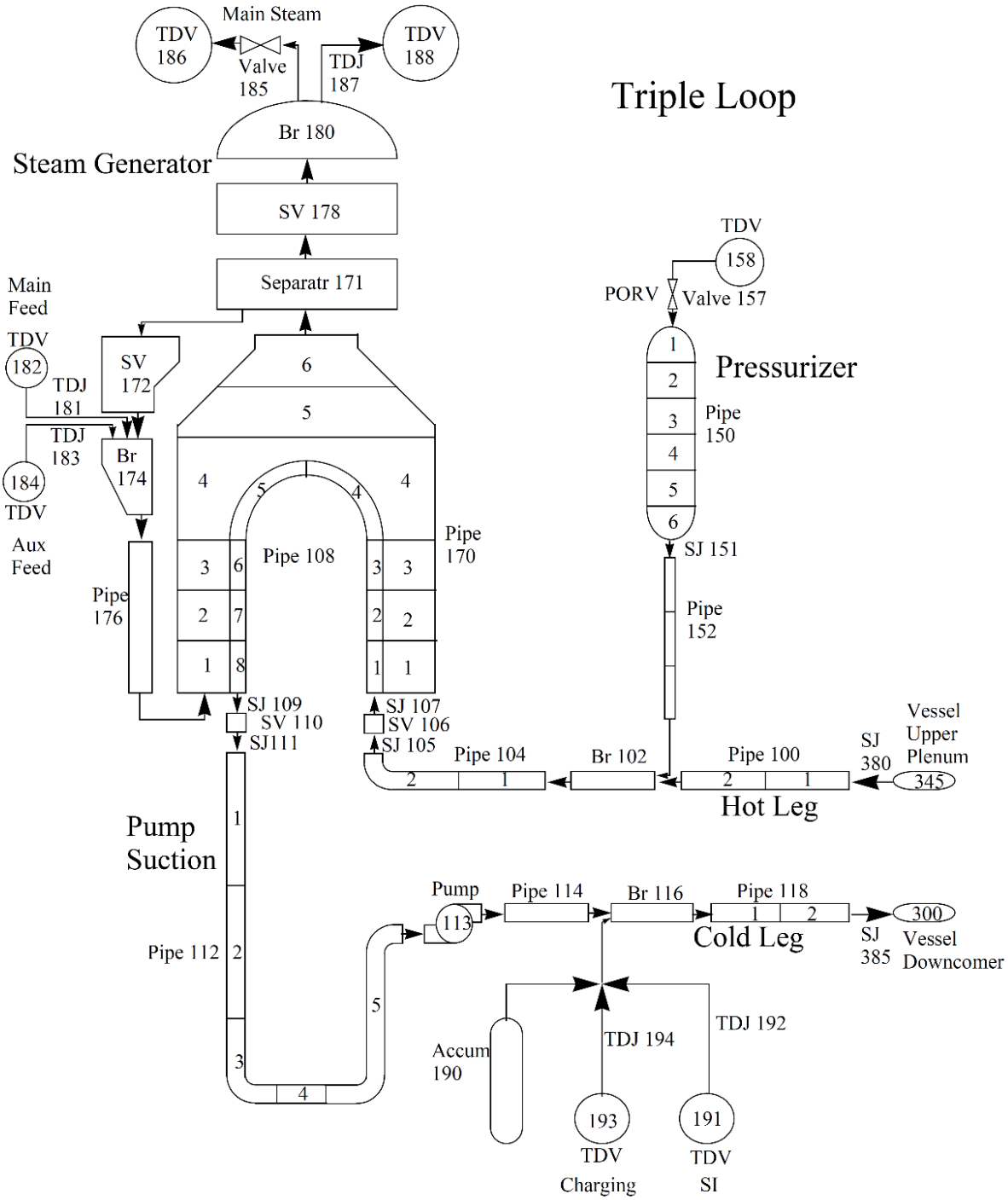

Figure 6.17 RELAP5 Nodalization for PWR: Loop Model (in Ref. [128])

In this model, the perturbed parameters and their standard deviation are listed in Table 8. Taking the pump degradation problem as an example, different from unknown FDI attacks, the transient reactor behavior of pump degradation is available via high fidelity model simulations or operation history, we identify the active subspace of three scenarios: low level, medium level and high level of pump degradation. Similarly, two signatures are constructed from the observation matrix, both of which are $1 \times n_o$ vectors, denoted as α_{H1} and α_{H2} for high level degradation pump, α_{M1} and α_{M2} for medium level degradation pump and α_{L1} and α_{L2} for low level degradation pump.

Table 8 Perturbed Parameters and Standard Deviation

	Perturbed Parameters	Normal_std
Initial Condition	Power level	0.05
	Inlet temperature	0.005
	Coolant temperature at core	0.005
Hydraulic Parameters	Loss coef in SG 1ry inlet	0.05
	Loss coef in SG 1ry outlet	0.05
	Loss coef in cold leg	0.05
	Loss coef in Prszr junction pipe	0.05
	Loss coef in SG 2ndary (loop 1)	0.05
	Loss coef in hot leg(1)	0.05
	Loss coef in hot leg(2)	0.05
	Loss coef in SG 1ry(1)	0.05
	Loss coef in SG 1ry(2)	0.05
	Loss coef in pump outlet	0.05
	Loss coef in SG 2ndary (loop 2)	0.05
	Loss coef in downcomer	0.05
	Loss coef in core	0.05

The pump and working fluid relationship is defined by empirically constructed curves relating to the volumetric flow and pump velocity of the pump head and torque. Pump characteristic curves, also referred to as four-quadrant curves, present the information in term of actual head, H, torque, τ, volumetric flow Q, and angular velocity ω, which are generally available from pump manufacturers. For used of RELAP5, the physical quantities like pump head representing characteristic curve need to be condensed to a ratio, resulting in new dimensionless versions of pump characteristic curve, denoted as homogenous curves. The construction of dimensionless curves require a series of rated physical quantities such that the actual head, H, can be nondimensionalized as $h = H/H_R$, where H_R represents the rate pump head. The same nondimensionalization of the rest pump parameters is calculated as: $\alpha = \frac{\omega}{\omega_R}$, $v = \frac{Q}{Q_R}$, $\beta = \frac{\tau}{\tau_R}$. The pump characteristic curves shown in Figure 6.18 represent one version of condensed characteristic pump curve, named as HVN, where x-axis is α/v, and y-axis is h/v^2. The blue, orange and grey curves represent the homogenous curves with low level, medium level and high level degradation, respectively and the corresponding normalized pump flow rate is plotted in Figure 6.19, in red line, blue line and green line.

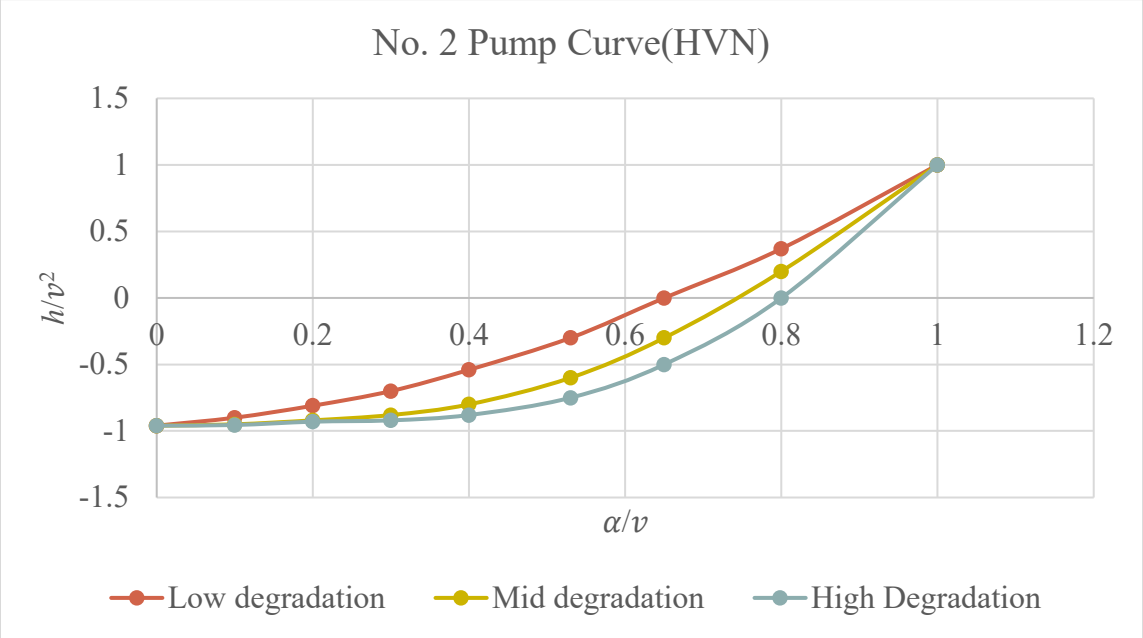

Figure 6.18 Pump characteristic curve

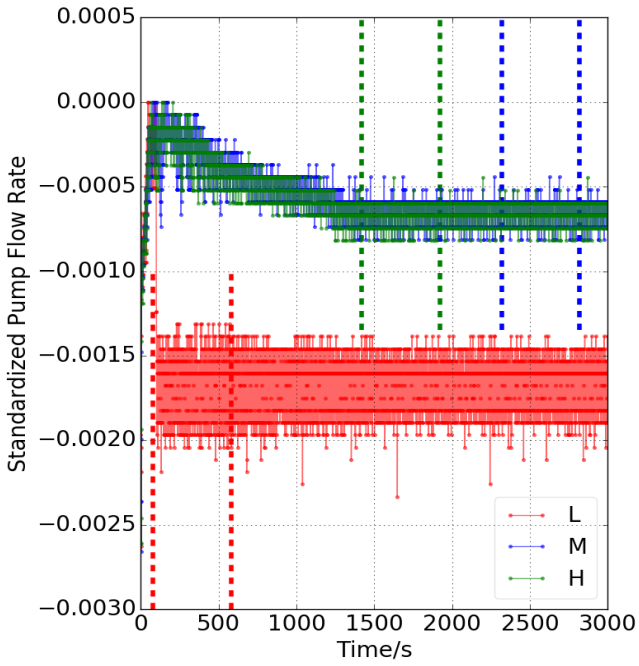

Figure 6.19 Normalized pump flow rate

6.3.2 Numerical Results

In this study, RWD is applied to distinguish between a high-level degradation scenario (denoted by "H" in the figures' title) and the cases with medium-level degradation (denoted by "M"). Figure 6.20 shows a scatterplot with classification of the selected HOC and LOC for the flow rate with high-level degradation and with medium-level pump degradation. For the pump flow rate using a time window of 200 seconds ("wd = 70"). The blue dots ("H") mark the behavior with high level degradation, and the red dots ("M") for data with medium degradation. Figure 6.20 condenses the time-evolution of the LOCs and HOCs components using a simple Euclidean norm, from which one can see that the data points from different pump degradation levels exist in different clusters such that even with small hyperparameters ($C = 0.01, \gamma = 1$), the classification accuracy reaches 99%. The result indicates the effectiveness of RWD for identification of different degradation level of a certain component.

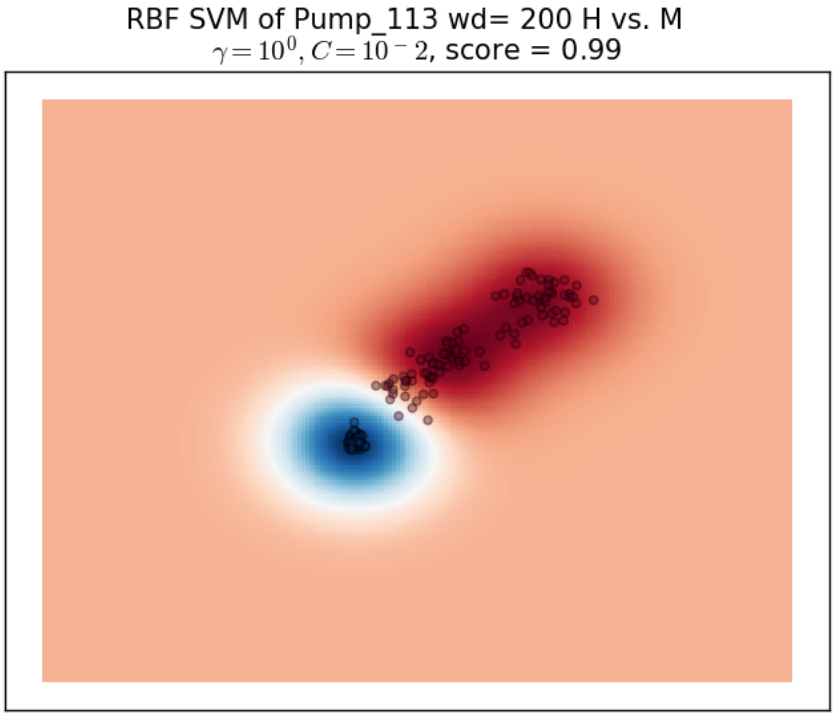

Figure 6.20 Classification Results for the Norm of the Feature of Steam Generation Amount

6.4 Results Summary

With the recent successful attempts against the digital control systems of critical infrastructures, there is a need to develop new defense strategies that recognize that state-sponsored attackers can ultimately gain access to the raw data used to control system behavior, and can falsify operational data in a manner that does not trigger conventional outlier/anomaly detection techniques in order to go undetected, which is referred to as false data injection attacks. Recent R&D efforts [12] promoting the use of model-based defenses offer a solution to this problem. This is achieved via the use of machine learning techniques to continuously compare the measurements from the real system with the measurements obtained from a physics-based simulation of the system in order to determine differences that may be indicative of false data injection attacks. A key assumption of such approaches is that the defender has the upper hand due to his/her sole access to the physics-based simulator. The preliminary study argues that for critical systems this may not be true because their design, operation, and safety, are all based on well-established practices, and their technical know-how is well understood. This work demonstrates that it is indeed difficult to develop a

predictive model for reactor behavior by relying solely on data-driven techniques, e.g., machine learning. However, with knowledge of the physics, it becomes possible to accurately learn the system behavior. In defense of model-based techniques, this work indicates that it was indeed difficult to employ an off-the-shelf inference capability to predict the true model parameters, and a complicated use of multiple techniques is elucidated, e.g., FFT, LS, ACE, and regularization. Based on this directional conclusion, this research explores the model-based defense and proposes a new OT defense to identify FDI attacks when the attacker has strong familiarity with the system, and has access to accurate models for dynamic system behavior. The idea is to rely on both dominant, referred to as the LOCs, as well as less dominant features, referred to as the HOCs, to derive signatures that can identify FDI attacks. This Chapter has helped introduce the basic idea and demonstrated its use to detect falsification in a single response using a single LOC and HOC components, and assuming the attack happens once over the time of the simulation. Results indicate the potential use of HOCs to build strong classifiers against FDI attacks which assume strong familiarity with the system. Future work will expand this work to develop LOCs and HOCs components across multiple responses, and will optimize the integration of HOCs components to maximize the classification accuracy.

7 EXPLORATORY STUDY II: REAL-TIME SUBTLE FDI DETECTION

The FDI detection in Chapter 6 is for the whole set of temporal measurements, aiming to validate the effectiveness and robustness of RWD approach. This approach is effective for offline data analysis, while in reality, the online monitoring system is adopted to track the evolution of the quantities of interest and assess the system state so as to ensure that nuclear safety-related facilities and instruments meet the objectives. However, the online monitoring data usually comes with significant noise. In Chapter 6, the 10% white noise has a small influence on the Euclidean-norm condensed signatures, but for online monitoring measurements without any condensing techniques, noise would hide the real information from further feature construction. Hence the denoising of the raw measurements is an essential preprocessing procedure. In addition, the components employed in this work are not the most dominant ones but with the intermediate strength to detect intrusions launched by well-resourced. Thus, normal denoising technique may not be qualified to smooth the noisy data without losing the information from the subtle variation. In this chapter, a multi-level denoising approach is proposed here to work with the FDI attack detection.

7.1 Mathematical Development

7.1.1 Denoising technique

Current denoising techniques can be briefly categorized into three types based on the underlying smoothing model: moving window based, locally regression-based and reduction-based. A commonly used window-based smoother is the moving average filer, which calculates a series of averages of sequential subsets of the full response profile. The mathematical expression can be found in Eq. (28), where the averaging window length is denoted as M, $y_j^{(0)}$ represents the j^{th} original noisy signal, $j \in [i, i+M]$ and $\overline{y}_i$ represents the i^{th} smoothed signal.

$$\overline{y}_i = \frac{1}{M}\sum_{i}^{i+M} y_j^{(0)}$$

(28)

While in the simple moving average the past observations are weighted equally, exponentially weighted moving average (EWMA) assigns exponentially decreasing weights over time; in other words, exponential smoothing assigns smaller weights for historic data and larger weights for recent data. Moving-window based denoising has variants like autoregressive integrated moving average (ARIMA), moving median, Gaussian-weighted moving average, central moving average, recursive moving average etc. These approaches are widely adopted due to their easy implementation and fast execution. In practice, however, a satisfying denoising result cannot be achieved without the setting of appropriate window size and the weights of the data. Regression-based smoothers were subsequently proposed to approximate segments of the data using polynomial functions. The basic idea of regression-based denoising is to find a proper regression model to estimate a segment of the whole profile, which combines the least squares regression with the flexibility of nonlinear regressions. Broadly speaking, regression-based smoothing encapsulates the moving window based denoising techniques, since the simple moving average is a special case of linear regression on each moving window. However, the regression based smoothing techniques solve the setting of weights via least-squares or the optimization of other loss functions. Examples of commonly used regression-based smoothing techniques include but are not limited to locally estimated scatterplot smoothing (LOESS), locally weighted scatterplot smoothing (LOWESS), spine smoothing etc. While the locally regression-based denoising techniques do not require physics insight to specify a global function of any form to fit a model to the data, this type of denoising techniques require a large of computational resources and increase the denoising cost. Another commonly used denoising technique, Kalman filtering, can be considered as a fast-implemented case of regression-based approach. Different from LOESS or its variants that the choice of regression model is more or less arbitrary, Kalman filter estimates the state of dynamic system behavior based on prior knowledge; in other words, the choice of the regression model depends on expertise of the dynamic model.

Whereas both moving-window based, and regression based denoising approaches function based on nearest or historical data, reduction-based approaches take advantage of the reduced complexity

of most models. Since noise typically represents the non-dominant/redundant aspect of a model, restricting the data to its dominant components often has a denoising effect. The reduction usually happens in temporal and frequency domain. In frequency domain, for example, Fourier transform decomposes a time series or an image into frequency components and the corresponding Fourier spectrum exhibits peaks for dominant frequency components. The reconstruction of the time series or image is based on removal of non-dominant frequency components. The same idea implemented in temporal domain can be represented by SVD, which decomposes the temporal snapshots into orthonormal vectors. An implementation of the reduction based denoising can be found in 7.1.1.1. The data denoising is accomplished by reconstruction of the original data solely depending on dominant components. Though the dominant behavior of the system is retained, the arbitrary removal of the components may remove part of information together with noise. Especially, the whole idea for the detection of subtle data falsification is based on its effect on the variation of HOCs, which requires delicate data preprocessing. To mitigate this dilemma, here a novel denoising approach is proposed, adopting a multilevel denoising approach to abstract more information and weed off more noise than a single execution of reduction approach.

7.1.1.1 Denoising Algorithm – Single Level

The evolution of a process variable is usually described as a certain state variable or output response as a function of time, here denoted as $\phi(t)$. While $\phi(t)$ can be expanded by different ways, like Taylor series expansion, fast fourier transform, matrix factorization etc., generally this expansion can be expressed in Eq. (29). In Eq. (29), j represents the jth samples/observation of the variable of interest. $\varphi_i(t)$ is the ith singular function (or vector) of a set of functions (or vectors) representing the first r DOFs (active) of the physics model. Also, they represent a mathematical basis for the active subspace. The α_i are the components of the function $\phi(t)$ along the active DOFs $\varphi_i(t)$.

$$\phi_j(t) \approx \sum_{i=1}^{r} \alpha_{i,j} \varphi_i(t) \tag{29}$$

For computational convenience, the active subspace basis functions $\varphi_i(t)$ are selected to be orthonormal, such that the components α_i can be readily calculated as inner products of the form:

$$\alpha_i = \varphi_i^T(t)\phi(t) \tag{30}$$

In doing so, the functions $\varphi_i(t)$ are selected such as to minimize the reduction error e_r, given by:

$$e_r = \min_{\varphi_i} \sum_{j=1}^{N} \left\| \phi_j(t) - \sum_{i=1}^{r} \alpha_{i,j}\varphi_i(t) \right\|^2 \tag{31}$$

Each one of the basis function is referred to as an active DOF, and collectively as the active subspace. The active DOFs are indexed from being most to least dominant, with the dominance measuring their contribution to the original function, i.e., the first active DOF is the most dominant such that variations in its associated coefficient α_j result in the most function variation in $\phi(t)$.

For monitoring applications, $\phi(t)$ is formed as a series of discrete time measurements, here denoted as a one-dimensional column vector with length n, mathematically expressed in Eq. (32), where y_i represents the measurement at i^{th} time step.

$$\phi(t) = \mathbf{y} = (y_1, y_2, \cdots, y_i, \cdots, y_n), i = 1, 2, \cdots, n \tag{32}$$

It is assumed that the analyst has access to many snapshots of a series of temporal evolution for reference, or normal function variations, either obtained from repeated model execution that simulates a plethora of normal operating conditions or from historical data, denoted here as:

$$\phi^{\text{Ref}}(t) = \mathbf{y}^{\text{Ref}} = \left[y_1^{\text{Ref}}, y_2^{\text{Ref}}, \cdots, y_i^{\text{Ref}}, \cdots, y_n^{\text{Ref}} \right] \tag{33}$$

These profiles are first normalized by their time-averaged values. No distinction between the normalized and original values will be made to avoid cluttering the notations.

Next, to get the active subspace basis functions $\varphi_i(t)$ in the context of discrete measurements, one relies on a window-based approach for monitoring, where a user-defined signature window of size

w is used to capture w-length snapshots of the vector of length n in Eq. (34). If sequential snapshots are taken, the vector in Eq. (34) is turned into a rectangular Hankel matrix, $\mathbf{H}^{\text{ref}}$, $w \times (n-w+1)$:

$$\mathbf{H}^{\text{Ref}} = \begin{bmatrix} \mathbf{h}_1^{\text{Ref}} & \mathbf{h}_2^{\text{Ref}} & \cdots & \mathbf{h}_{n-w+1}^{\text{Ref}} \end{bmatrix} = \begin{bmatrix} y_1^{\text{Ref}} & y_2^{\text{Ref}} & \cdots & y_{n-w+1}^{\text{Ref}} \\ y_2^{\text{Ref}} & y_3^{\text{Ref}} & \cdots & y_{n-w+2}^{\text{Ref}} \\ \vdots & \vdots & \ddots & \vdots \\ y_w^{\text{Ref}} & y_{w+1}^{\text{Ref}} & \cdots & y_n^{\text{Ref}} \end{bmatrix} \tag{34}$$

Mathematically, the SVD-based reduction of $\mathbf{H}^{\text{Ref}}$ may be described as follows:

$$\mathbf{H}^{\text{Ref}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\text{T}} \approx \sum_{k=1}^{r} s_k \mathbf{u}_k \mathbf{v}_k^{\text{T}} \tag{35}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r)$, $\mathbf{S} = diag\{s_1, s_2, \cdots, s_w\}$, $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r)$, and r represents the rank of the Hankel matrix. The column vectors in $\mathbf{U}$ matrix form a set of orthonormal bases for the column space spanned by $\mathbf{h}_k^{\text{ref}}, k = 1, 2, \cdots, r$, where r represents the rank of the Hankel matrix. The row vectors in $\mathbf{V}^{\text{T}}$ form a set of orthogonal bases for of the row space spanned by the rows of $\mathbf{H}^{\text{Ref}}$. $\mathbf{S}$ is a diagonal matrix storing singular values in a descending order. The rank of the matrix is usually determined by a user-defined tolerance, as expressed in Eq. (31). Then the active subspace basis functions $\varphi_i(t)$ can be expressed by the column vectors in $\mathbf{U}$ matrix, i.e. $\varphi_i(t) = \mathbf{u}_i$.

Denote a normalized temporal evolution from raw sensor readings (with noise) of the process variables as $\mathbf{y} = (y_1, y_2, \cdots, y_i, \cdots, y_n)$, $i = 1, 2, \cdots, n$ and the corresponding Hankel matrix as $\mathbf{H}_i = \begin{bmatrix} \mathbf{h}_{i-w+1} & \mathbf{h}_{i-w+2} & \cdots & \mathbf{h}_i \end{bmatrix}$, with w taken as window size, where i represents the i^{th} simulated/observed temporal measurements. The denoised measurements with single level reduction can be implemented via:

$$\begin{aligned} \hat{\mathbf{h}}_{i-w+} &= \mathbf{U}\mathbf{U}^{\text{T}}\mathbf{h}_{i-w+1} \\ \hat{\mathbf{h}}_{i-w+2} &= \mathbf{U}\mathbf{U}^{\text{T}}\mathbf{h}_{i-w+2} \\ &\vdots \\ \hat{\mathbf{h}}_i &= \mathbf{U}\mathbf{U}^{\text{T}}\mathbf{h}_i \end{aligned} \tag{36}$$

Therefore, the corresponding denoised Hankel matrix is expressed as:

$$\hat{\mathbf{H}}_i = \left[\hat{\mathbf{h}}_{i-w+1}, \hat{\mathbf{h}}_{i-w+2}, \cdots \hat{\mathbf{h}}_i\right] = \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{H}_i \tag{37}$$

Since one data point can be smoothed a maximum of w times, a given point y_i is no longer smoothed after i time steps. The final smoothed value, $\hat{y}_i$, is obtained by the first entry of $\hat{\mathbf{H}}_i$.

In the context of FDI detection, it is noteworthy that if a FDI attack vector δ is along a direction of the null space of $\mathbf{U}$, then this attack vector will be smoothed out after the adoption of the reduction-based denoising approach, which ties into the main drawbacks of this denoising technique.

7.1.1.2 Denoising Algorithm –Multilevel Approach

The multi-level approach can be thought of as simply extending the range of the smoothing function, i.e. the final smoothed point $\hat{y}_i$ is dependent on the points, ranging from $(i\text{-}w\text{-}v)^{\text{th}}$ to $(i+w+v\text{-}1)^{\text{th}}$, where v depends on the number of levels and their respective window sizes. While implementing multilevel denoising approach, one will get a final denoised data point after (w+v-1) time steps. Here, taking a two level denoising for demonstration, the implementation process is stated as below.

If the **u** vectors can be thought of as a polynomial approximation to the original curve, the multi-level **u** vectors are a good approximation of their components, components of their components, and so on. The basic idea of multilevel approach is to find sets of bases for each layer of denoising, the first of which is obtained as **U** matrix in section 7.1.1.1. Then, one needs to identify the basis for the second level variations that are stored in the projection matrix $\boldsymbol{\alpha}^{\text{ref}}$.

$$\boldsymbol{\alpha}^{\text{ref}} = \mathbf{U}^{\alpha\mathrm{T}}\mathbf{H}^{\text{ref}} \tag{38}$$

Similarly, SVD is adopted to identify the basis of the second level variations, as shown in Eq.(39), where the superscript α represents the second level mapped variation related quantities/matrix and the rank of the second level variations spanned space is denoted as *s*.

$$\boldsymbol{\alpha}^{\text{ref}} = \mathbf{U}^{\alpha}\mathbf{S}^{\alpha}\mathbf{V}^{\alpha\text{T}} \approx \sum_{k=1}^{s} s_k^{\alpha}\mathbf{u}_k^{\alpha}\mathbf{v}_k^{\alpha\text{T}} \tag{39}$$

To capture the main variation of the process variable's components, i.e., HOCs, one needs to project the temporal measurements onto the **U** matrix to capture most dominant variations for current measurements, mathematically, which can be expressed in Eq. (40). Speaking in a context of online monitoring, a dynamic Hankel matrix $\mathbf{H}^i$ is constructed with the window size for the second level components, where the superscript *i* of $\mathbf{H}^i$ represents the index of the data points.

$$\boldsymbol{\alpha}^i = \mathbf{U}^{\text{T}}\mathbf{H}^i \tag{40}$$

$$\mathbf{H}^i = \begin{bmatrix} \mathbf{h}_{i-v+1} & \mathbf{h}_{i-v+2} & \cdots & \mathbf{h}_i \end{bmatrix} \tag{41}$$

The multilevel denoising works as inner iteration for the high-level variation mapped matrices. Here the denoised second level variations is expressed as:

$$\hat{\boldsymbol{\alpha}}^i = \mathbf{U}^{\alpha}\mathbf{U}^{\alpha\text{T}}\boldsymbol{\alpha}^i \tag{42}$$

With the denoised high-level variations, the low-level variations can be recovered by being mapped back to the basis at the corresponding level, as shown in Eq. (43):

$$\hat{\mathbf{H}}^i = \mathbf{U}\mathbf{U}^{\text{T}}\hat{\boldsymbol{\alpha}}^i \tag{43}$$

Therefore, the eventually smoothed data points corresponded Hankel matrix can be expressed in Eq. (44).

$$\hat{\mathbf{H}}^i = \mathbf{U}\mathbf{U}^{\alpha}\mathbf{U}^{\alpha\text{T}}\mathbf{U}^{\text{T}}\mathbf{H}^i \tag{44}$$

As stated on page 116, the final smoothed measurements are stored in the first entry of the updated $\hat{\mathbf{H}}^i$ matrix, since the measurement y^i is the last entry of the matrix $\mathbf{H}^{i-w+1}$ and the first entry of the matrix $\mathbf{H}^{i+w-1}$. Similar to the idea of central difference in the numerical methods, the updating process of y^i is related to ($2w$-1) data points, from y^{i-w+1} to y^{i+w-1}. Expand this scheme to multilevel denoising technique, there are [2(w+v) -1] data points are involved in the y^i updating process. In other words, the system will get the final smoothed data after [(w+v)-1] time steps.

For illustration, overall denoising calculation scheme can be found as below:

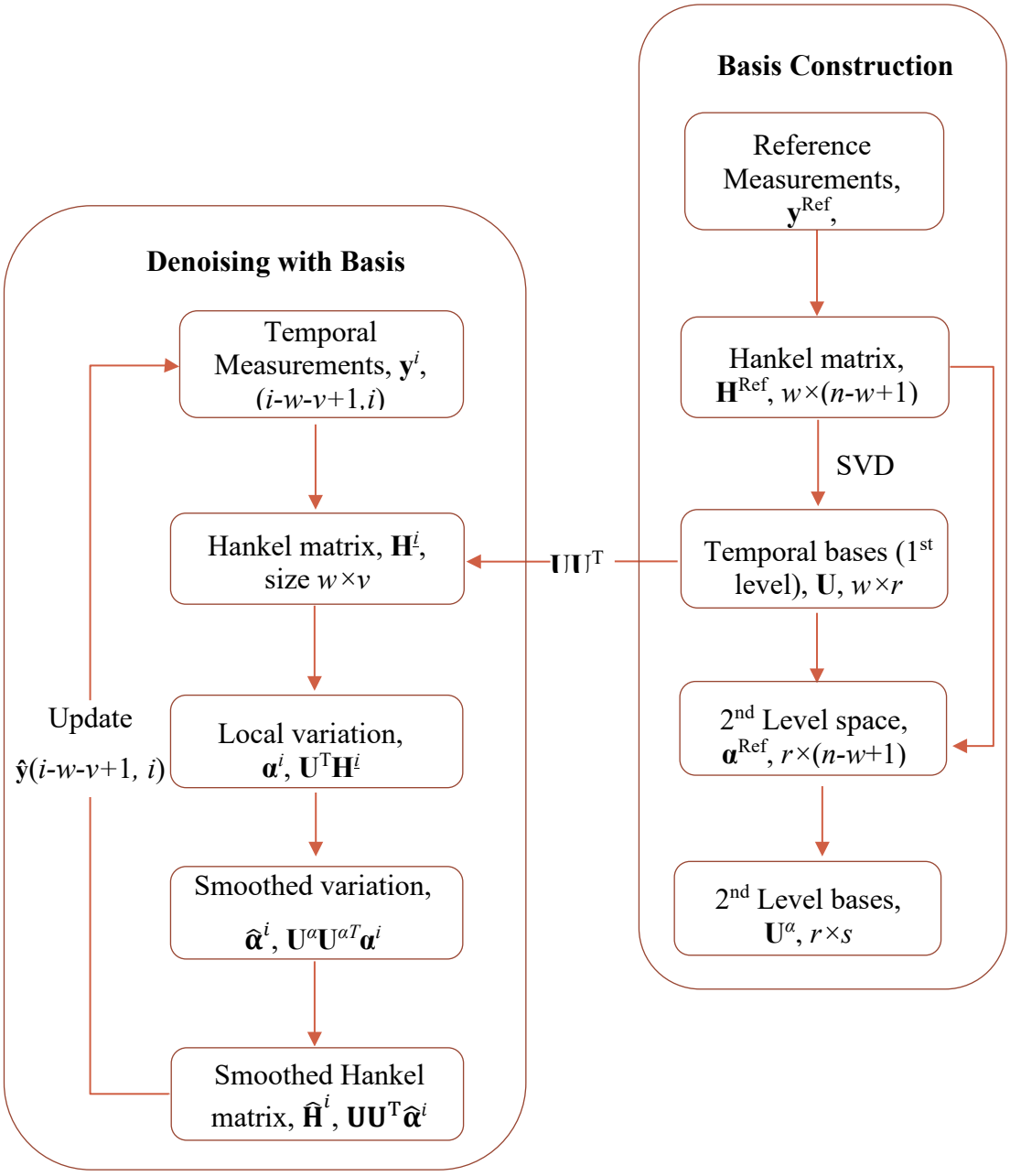

Figure 7.1 Multilevel denoising calculational scheme

Understanding the basic idea of the multilevel denoising approach, one can easily expand this algorithm to more level denoising, which can be expressed as:

$$\hat{\mathbf{H}}^{i} = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3\cdots\mathbf{U}_m\mathbf{U}_m^{T}\cdots\mathbf{U}_3^{T}\mathbf{U}_2^{T}\mathbf{U}_1^{T}\mathbf{H}^{i} \tag{45}$$

where m represents the total number of layers adopted for denoising. Also, the final smoothed y^j will be obtained after T time steps, expressed as in Eq. (46), where w_l represents the window size at layer l.

$$T = \sum_{l=1}^{m} w_l - 1 \tag{46}$$

As one may intuitively think, the number of layers, m, will not increase to infinity, since the information carried is decreasing as the number of layers increases. Thus, to implement the multilevel denoising approach, one needs to conduct a series of tests for a proper selection of the number of layers and rank identification of each layer.

While this multilevel denoising technique is robust for it only requires the measurements data of process variables instead of physics insights or state estimation, a generic theoretical limit for the number of layers is undeveloped.

7.1.2 RWD Equipped with Multilevel Denoising for Online Monitoring –Single Process Variable

In this section, RWD is implemented with aid of multilevel denoising to fulfill the online detection of triangle FDI attack. The basic idea of RWD stated in 6.1.2 is to track the variation of the relationship between LOCs and HOCs in online monitoring. The discrepancy of the relationship between genuine data and the online data will issue an alarm. Adopting the same set of notations of the denoising section 7.1.1.2, the steps of RWD for online monitoring are stated below:

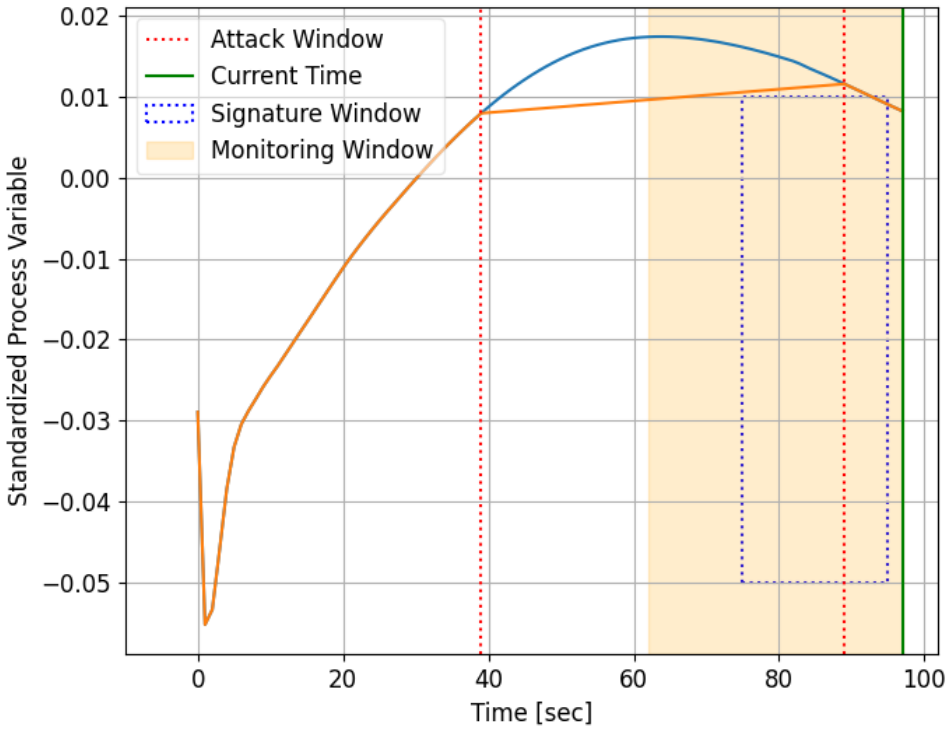

Figure 7.2 Illustration for Sliding Signature Window

1. Employing a signature window of size w, sequentially place this window over the denoised genuine signal $\hat{\mathbf{y}}^g = \left(y^g_{i-w-v+1}, \cdots, y^g_i\right), i = w+v, w+v+1, \cdots, n$, within the monitoring window, to generate v random snapshots for the window values, and aggregate in a matrix $\mathbf{H}^i$ ($w \times v$). The illustration of the sliding signature window within the monitoring window can be found in Figure 7.2.
2. Based on a defined tolerance, determine the rank, i.e., the number of components, denoted as r_{L} and r_{H} for LOCs and HOCs respectively. The LOCs, and HOCs are captured in two matrices $\mathbf{U}_{\mathrm{L}}$ ($w \times r_{\mathrm{L}}$), and $\mathbf{U}_{\mathrm{H}}$ ($w \times r_{\mathrm{H}}$).
3. Calculate the projection of each of the v windows from the $\mathbf{H}^i$ matrix along the r_{L} LOCs and the r_{H} HOCs as features and aggregate the features in two matrices of sizes $r_{\mathrm{H}} \times v$ and $r_{\mathrm{L}} \times v$, denoted respectively as $\boldsymbol{\alpha}^g_H$ and $\boldsymbol{\alpha}^g_L$.

4. The above steps are repeated for the attack signals $\hat{\mathbf{y}}^a = \left(y_{i-w-v+1}, \cdots, y_i\right)$, $i = w+v, w+v+1, \cdots, n$ to generate matrices $\boldsymbol{\alpha}_H^a$ and $\boldsymbol{\alpha}_L^a$, in order to detect attacks injected during the period from timestep i to timestep $i+w+v-1$. The attack signals are generated by placing windows randomly throughout the genuine data and replacing the values by linear piece-wise trends. Details on this may be found in a previous publication [42] and in the subsection 7.2.
5. Input datasets are prepared as a vector via the concatenation of LOC and HOC.
6. Label the feature, column vectors in in $\boldsymbol{\alpha}_H^g$, $\boldsymbol{\alpha}_L^g$, $\boldsymbol{\alpha}_H^a$ and $\boldsymbol{\alpha}_L^a$ that do not contain falsified data as '0' and the other ones as '1' for containing altered data.
7. Train a binary support vector machine (SVM) classifier to identify the feature vectors constructed from attack data.

For illustration, the calculational scheme of the algorithm is shown in Figure 7.3.

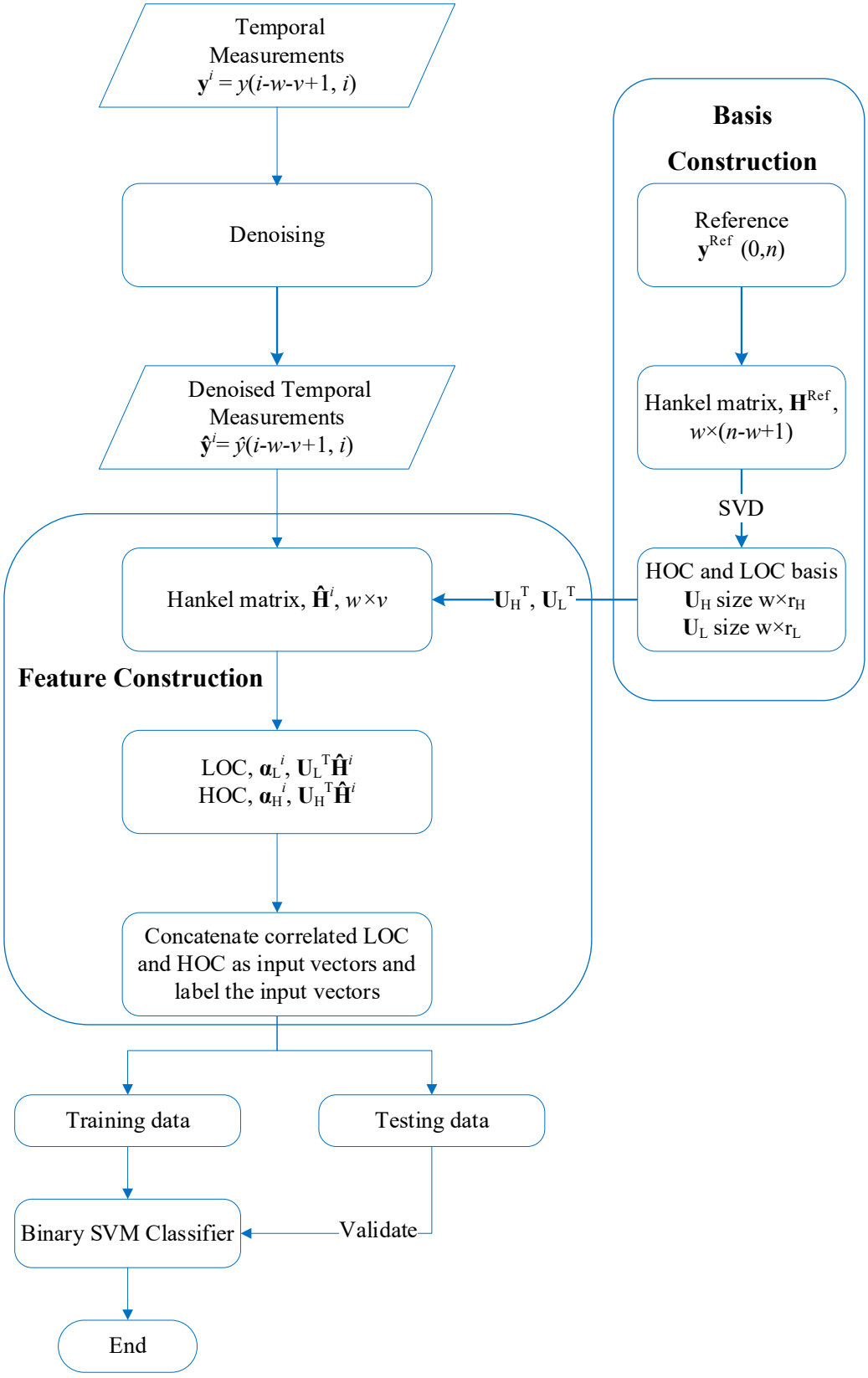

Figure 7.3 Calculational Scheme of Detection Algorithm –Single Process variable

7.1.3 RWD Equipped with Multilevel Denoising for Online Monitoring –Multi Process Variables

Besides employing LOCs and HOCs from one process variable, another option is to retrieve information from different process variables, especially the correlated ones. As one can think intuitively, while conducting online monitoring, the relationship between components from different process variables contains two sources of information: autocorrelation and cross-correlation between the process variables. This allows one to detect attack scenarios in which the attacker has access to some of the historical sensor's data, allowing them to perform a reply attack. To simulate this, we assume that some of the responses are duplicated from historical data, and the rest are falsified by the attacker. We show that the combined use of LOCs and HOCs allows for the detection of this attack scenario. To facilitate the FDI detection algorithm with the cross correlation, here we expand the algorithm in 7.1.2 to another version involving multi process variables.

The basic idea to employing the components of multi process variables focuses on the construction of the input vector for classifying. Specifically, the input vectors contain the HOC or LOC information from the different process variables. However, the construction of input vector via arbitrarily stacking of components will weaken the classifier performance or add calculational burden to the classifier. To mitigate these issues, a pre-analysis of the components of different variables is necessary. Taking two process variables as an example, with a user-defined tolerance, one can obtain the active DOFs for both variables, denoted as r_1 and r_2. Then one can build a $r_1 \times r_2$ correlation matrix, from which one can identify the component pairs with the least uncertainties. The identified pairs can be employed to construct input vectors. For illustration, the calculation scheme of the algorithm with multi process variables can be found in Figure 7.4, where the subscript 1 and 2 represent different process variables and the hat notation represents the denoised values. Also, to avoid a verbose notation, the temporal index i for both temporal profile $\mathbf{y}$ and constructed matrix $\mathbf{H}$ is neglected.

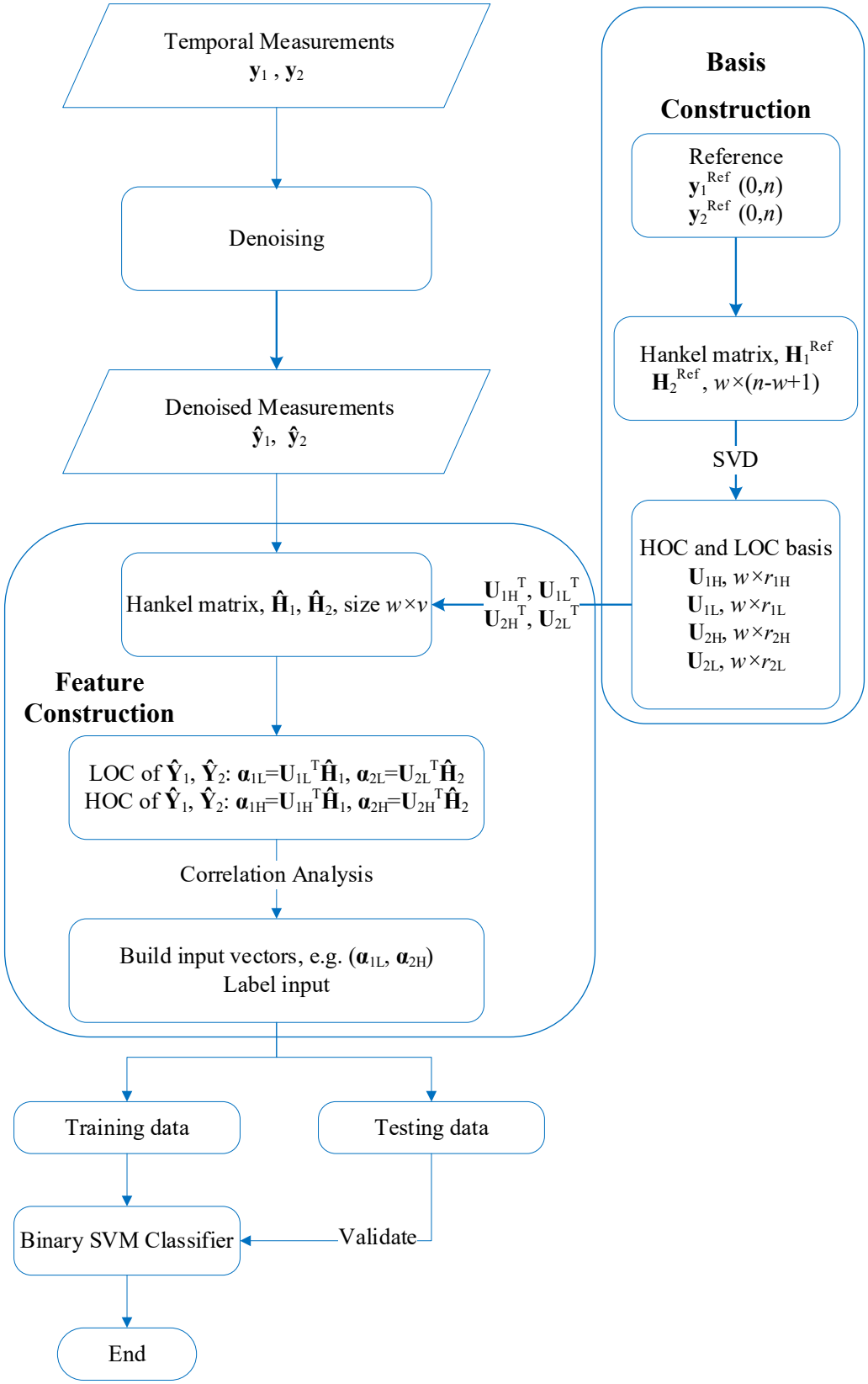

Figure 7.4 Calculational Scheme of Detection Algorithm –Multi Process variables

7.1.4 Evaluation of attack detection

This section defines a detection criterion based on the results of the SVM classifier. The raw SVM results provide information on how often the classifier is triggered. To quantify that over the temporal horizon, two metrics are defined, GC and AC. The GC, short for Genuine Coverage, measures the number of times the classifier returns a label of '0', denoting the genuine behavior, and the AC, short for Attack Coverage, measures the number of times the classifiers return a label of '1', denoting the attack behavior. In ideal settings, the classifier is expected to be triggered when the monitoring window is overlapping with the attack window.

As would be expected, if the overlap is small, the likelihood of the classifier being triggered will be lower than if the overlap is large. To minimize the rate of false positives, a criterion should be developed in terms of the AC and GC metrics. Both metrics are normalized by the total number of time steps in which the monitoring window has overlap with the attack window. For example, for an attack window of 50 seconds, a monitoring window of 25 seconds and a signature window of 20 seconds, there should be a total of 74 seconds in which the two windows overlap. Recall that the monitoring window is advancing one second at a time. Thus, a 20% AC implies that the classifier triggered a '1' label for 20% of the 74 seconds. These two metrics are used to determine a criterion for detection as follows. An attack is declared when the classifier triggers a '1' label five times in a row. This basic criterion is used in this work, however more complicated criteria may be used that take into account the score of each label, which is a function of the distance from the decision boundary of the classifier. Some of these ideas will be explored in future work.

In addition to the binary decision of attacking detection, it is important to determine the time-delay between the onset of the attack and its detection. Hence, a time delay, t_d, is defined in the context of real-time monitoring in Eq. (47), where t_p refers to the time step at which the classifier is triggered when the classifier declares positive predictions that last t time steps, and t_a refers to when the attack is injected. Ideally, one would want the detection time delay to approach zero.

$$t_d = t_p - t_a \tag{47}$$

7.1.5 Limits exploration

This manuscript focuses on the detection of subtle data falsification for online monitoring of nuclear system, which raises the question: how subtle can an attack be and still be detected? To explore this limit, a distance metric is defined measuring the discrepancy between the genuine and attack values over the attack window. This distance is defined in Eq. (48) below where y_l^g and y_l^a refer to the value of normalized raw genuine and attack response values at l^{th} time step, respectively, and w_a represents the size of the attack window. Our goal is to find the minimum value of d_l below which an attack may become indistinguishable from genuine data. Since a supervised learning setting is employed for the classifier, we use a threshold of $d = 0.35\%$, such that any deviation below that is not considered to be an attack.

$$d = \sqrt{\frac{\sum_{l=1}^{w_a} (y_l^g - y_l^a)^2}{w_a}} \tag{48}$$

7.2 Numerical Results

This section exemplifies the application of the proposed OT defense using a virtual approach, wherein real measurements are simulated using a dynamical reactor model based on the RELAP5 code with noise added to emulate real data collected in a nuclear reactor. For demonstration purposes, we focus solely on triangle attacks, proposed recently as a simple yet effective form for evading detection by conventional data-driven and/or model-based OT defenses; a short overview of triangle attacks is given in the next subsection. Both univariate and multivariate monitoring renditions of the proposed OT defense are demonstrated in the following subsections.

7.2.1 Subtle data falsification: triangle attack

The basic idea of triangle attack is that given a sequence of measurements, i.e., signal values of a process variable containing noise, a series of line segments are calculated to adjust the trend of the process variable variations with artifact noise added to emulate the noise in the raw data before they are falsified. Calculation of these line segments does not require knowledge about the

dynamical model governing the system behavior, as it employs simple rules to find the best linear trends, then superimposes noise that is consistent with the noise in the raw data as illustrated in Figure 7.5. The red noisy data set represents the raw measurements, and the black line segments represent the calculated trends. The falsified data represent the sum of the linear trends, to be selected by the attacker to change the system state, and artifact noise designed to evade replay attack detection. The detailed calculational procedures of the triangle attack may be found in [43].

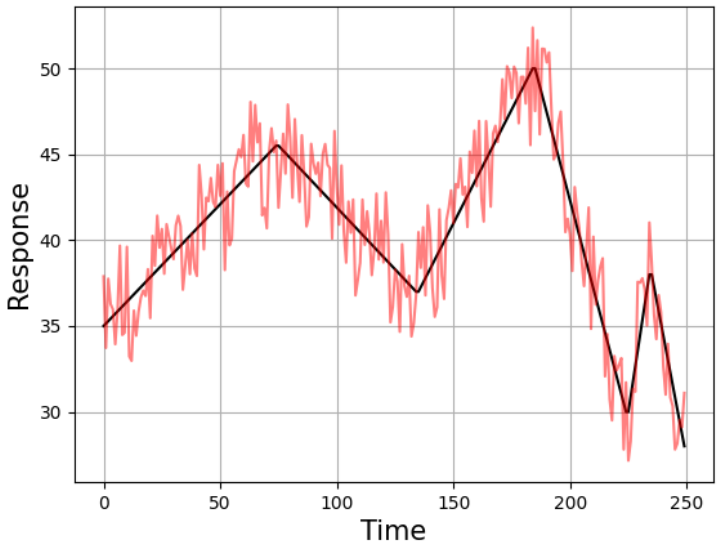

Figure 7.5 Line Segments Fit of Triangle Attack

7.2.2 Denoising results

In this subsection, multilevel denoising approach is applied to preprocess the online monitoring data. For demonstration, moving window average denoising and a single-level denoising techniques are employed to compare with the multilevel denoising approach, where the single-level denoising approach share the same idea with the current SVD denoising approach. In Figure 7.6, the grey dots in all three subplots represent the original noisy measurements of the pressure in the secondary side of steam generator, which is generated via a 10% white noise addition onto the simulated noise-free response profile, shown as the purple line in the last subplot. The blue line in the first subplot represents the smoothing result of EMWA; the green line in the second subplot shows the denoising temporal evolution of pressure; the multi-level denoising temporal profile is

plotted as red line in the last subplot. Seeing from the first two subplots, one can tell that there are some tiny wiggles shown on both green line and blue line, while the multilevel denoising approach smooths out the tiny wiggles as shown in the third subplot, which are not represented by the non-noisy profile. Here the number of components employed for single level denoising is determined by a user-defined tolerance shown in Eq.(29). Similarly, the number of components employed in the second level denoising is also determined by a user-specified denoising optimal. Here the window size, w, to build the Hankel matrix for the reduction-based smoother is same with the one for EMWA smoother, 20.

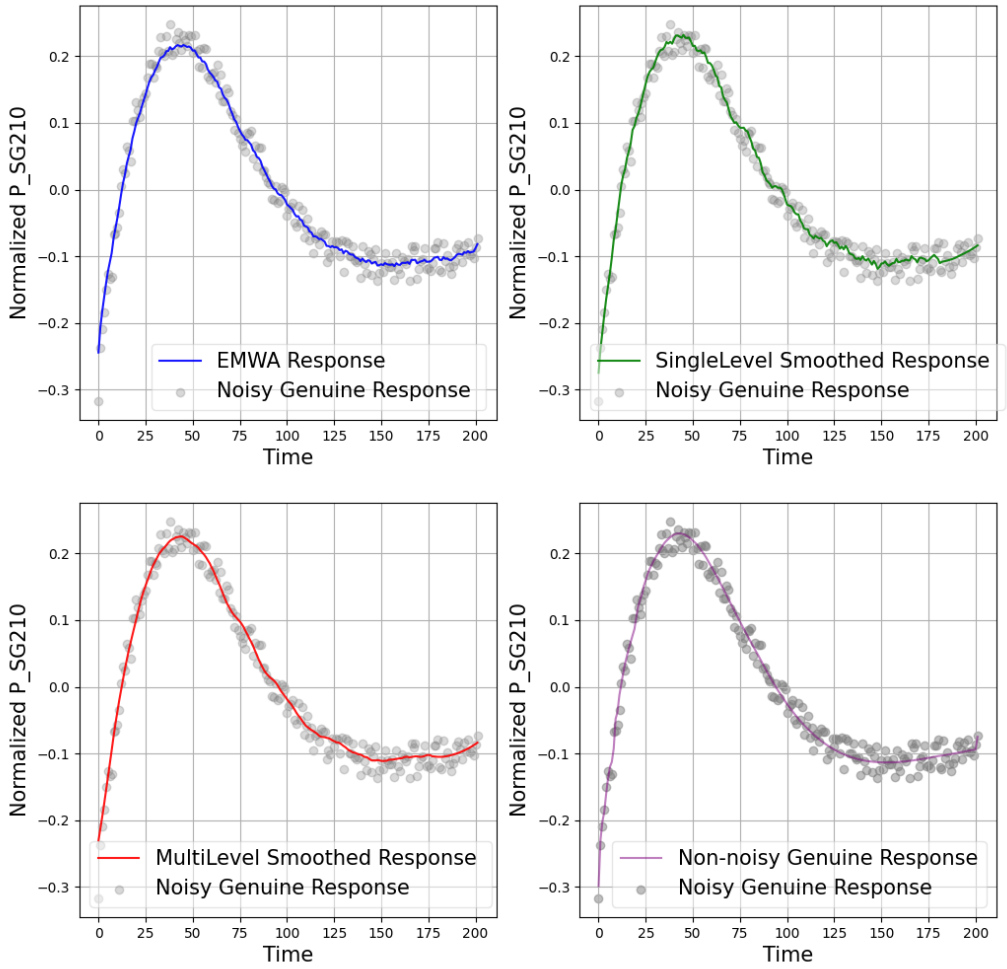

Figure 7.6 Denoising Results Comparison

From As aforementioned, a key contribution of this paper is the design of a denoising algorithm to ensure that subtle variations can be distinguished from the noise. To measure the impact of the

noise, two signal-to-noise ratio type metrics are employed, defined as the ratio of the noisy signal to the noise within the attack region, mathematically expressed as S_{noisy}/N; another metric is employed as a measure of the meaningful signal, defined as the ratio of the noise free signal to the noise within the attack window, mathematically expressed as S_{clean}/N, where S_{noisy}, S_{clean}, and N are defined in Eq. (49). In our context, the both signals refer to the difference between the genuine and falsified data, where the subscripts g and a refer to genuine signals and attacked signals, respectively; the subscript l refers to the temporal index of the profiles; the attack window length is denoted as w_a. The noise is defined as the discrepancy between the raw sensor readings y_l^g and the simulated signals $y_l^{g_0}$,where the subscript "0" represents noise-free profiles. To construct the subtle attack vector, the meaningful signal, i.e., S_{clean} should match the noise level or even be smaller.

$$
\begin{aligned}
S_{\text{noisy}} &= \sqrt{\frac{\sum_{l=1}^{w_a}(y_l^g - y_l^a)^2}{w_a}} \text{ over the attack window} \\
S_{\text{clean}} &= \sqrt{\frac{\sum_{l=1}^{w_a}(y_l^{g_0} - y_l^{a_0})^2}{w_a}} \text{ over the attack window} \\
N &= \sqrt{\frac{\sum_{l=1}^{w_a}(y_l^{g_0} - y_l^{g})^2}{w_a}} \text{ over the attack window}
\end{aligned}
\tag{49}
$$

7.2.3 FDI Detection with Univariate Monitoring

This subsection applies the OT defense to a single monitored process variable, using a two-level denoising methodology. For the first level, the first two column vectors in $\mathbf{U}_1$ in Eq.(45) are used per Eq. (37), and for the second level, the first column vector in $\mathbf{U}_2$. The window size for the temporal level, w, is taken as 20, and the window size for the components level, v, is 10. The time delay resulting from denoising is 29 timesteps based on Eq. (46).

The results are shown in the subplots in Figure 7.7 to Figure 7.9. In the first subplot, the noisy genuine and attack temporal profiles of the monitored process variable, the temperature at the primary side of SG, are shown as green and red dashed lines separately, while the non-noisy genuine and attack temporal evolutions are respectively plotted in blue and orange solid lines, respectively. The added noise profile follows a normal distribution with mean value as 0 and standard deviation as 0.2%. The noisy profile is calculated via Eq. (50), where y_i denotes for noisy sensor readings, n_i denotes for the added noise, and y_i^0 denotes for the noise-free/simulated variable, the subscript i refers to the index of time step.

$$y_i = (1 + n_i) \cdot y_i^0 \tag{50}$$

The second subplot shows the denoised temporal evolution of the monitored process variable. The noisy profiles are also represented in here for a complete results demonstration. The third subplot contains the relationship between the LOC and the HOC features extracted from genuine and attack signals. Here the genuine datasets are labelled as '0' and the others containing falsified data are denoted as '1'. In the fourth subplot, the blue horizontal dots show the true labels of genuine input datasets as a function of time, and the orange dots show the labels of attack datasets. Ideally, for a given genuine profile, the OT defense should issue a '0' label for all the time step, and for an attack profile, it should issue a '1' label over the attack window, whose onset is marked by a vertical red solid line. The fifth subplot shows the prediction of the OT defense when a genuine profile is applied, i.e., all predicted labels are '0' as would be expected. The sixth subplot shows the results when the OT defense is presented with an attack profile. Results shows that a label '1' is declared 0.0 ~ 50.0 seconds after the onset of the attack, and the cases with delay not less than 50 seconds are considered as undetected.

With the premise stated in subsection Limits exploration7.1.5, here the alarming time limit is selected as 5 seconds and the difference limit expressed in Eq. (48) is chosen as 0.35%. In other words, when the noisy signal triples the noise, the attack is considered as a valid attack, based on which if the predicted alarming time lasts over 5 seconds, the attack can be considered as detected. The detection performance is evaluated by AC, GC and detection delay, which are shown in

subtitles of the fifth and sixth subplots. The sample index and the corresponding signal to noise ratio, S_{clean}/N are represented in the figure title.

The simulation has been executed 1000 times, all representing a range of operational conditions, achieved by randomly sampling initial and boundary conditions as well as some of the model's parameters. A total of 750 samples were used to train the classifier, and the remaining 250 samples were used for testing. For each of the training sample, an attack window is randomly placed, and the trend is changed to piecewise linear. Noise is added to both the original values (representing genuine behavior) and the attack values. Based on the position of attack windows, we select for demonstration three different attack regions: a region where the given response is increasing (Figure 7.7), decreasing (Figure 7.9), and in-between where a peak is expected (Figure 7.8). These three regions will be denoted by the "increasing", "decreasing", and "peak" regions, respectively.

Figure 7.7 shows an example of FDI detection results using the developed multilevel denoising method, in which the attack is injected in the increasing region. Analysis of the relationship between HOC and LOC in the second subplot reveals that a small difference in the temporal evolution of a response could lead to a significant difference in the HOC-LOC pattern, based on which a classifier can be reliably trained. The classifier predicted labels for genuine data are shown in the fifth subplot, all as '0', indicating that the classifier will not misclassify the genuine data as attacked data and issue a false alarm. In the last subplot, the predicted labels for attack data represent as '1' without a time delay. The bottom left subplot shows that the classifier is triggered at 75.6% of the time when the monitoring and attack window overlapped, i.e., AC = 0.756. Based on the 5 second detection criterion, this attack is hence detectable. Similar results for the peak and decreasing regions are shown in Figure 7.8 and Figure 7.9, respectively.

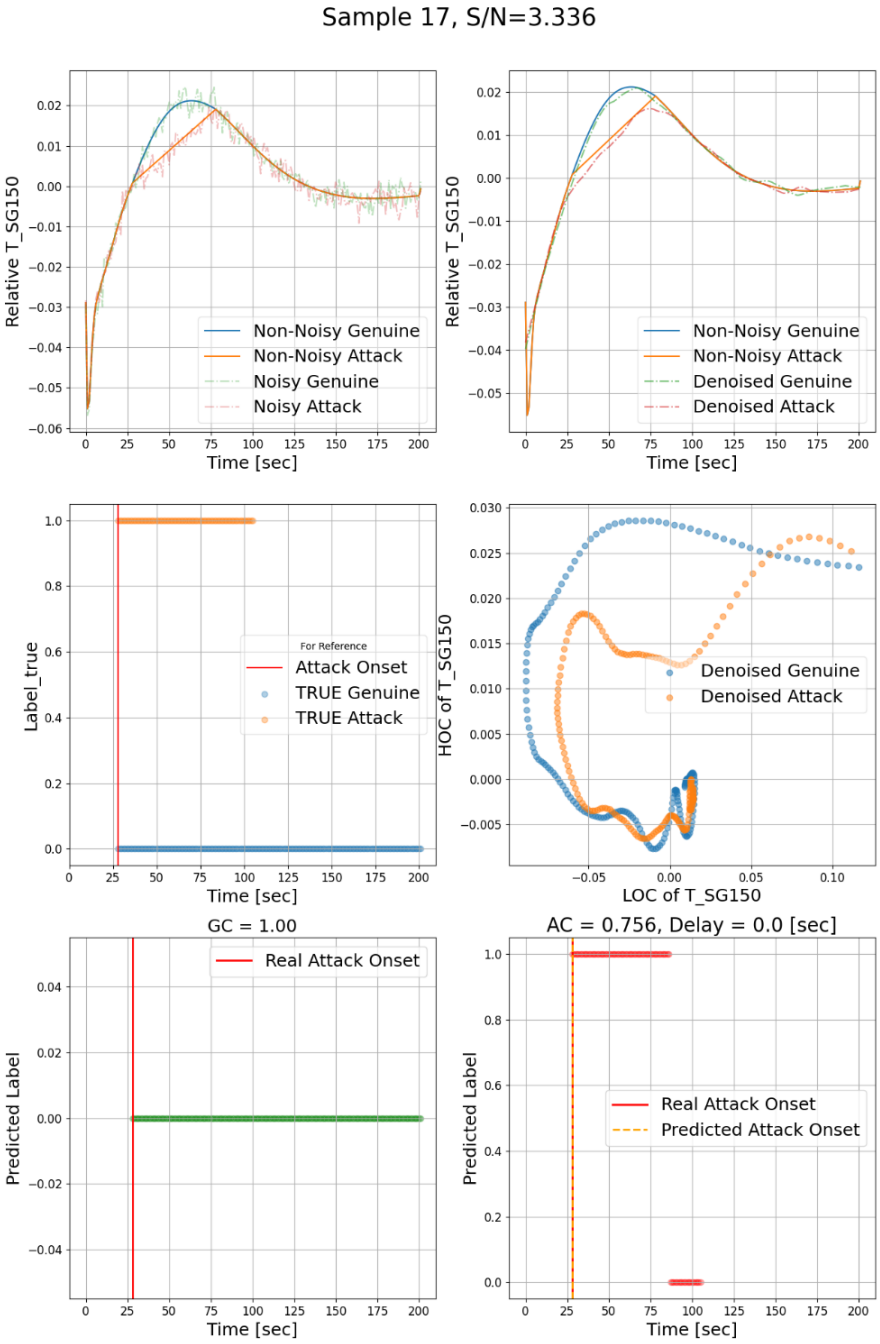

Figure 7.7 FDI detection with Multilevel denoising (Region 1)

Figure 7.8 and Figure 7.9 show the results where the attack vector in injected in a quasi-steady region. While the RWD algorithm supported with single level denoising technique cannot detect the FDI attack, the multilevel denoising technique is capable of the detection with a delay time equals to 11 seconds.

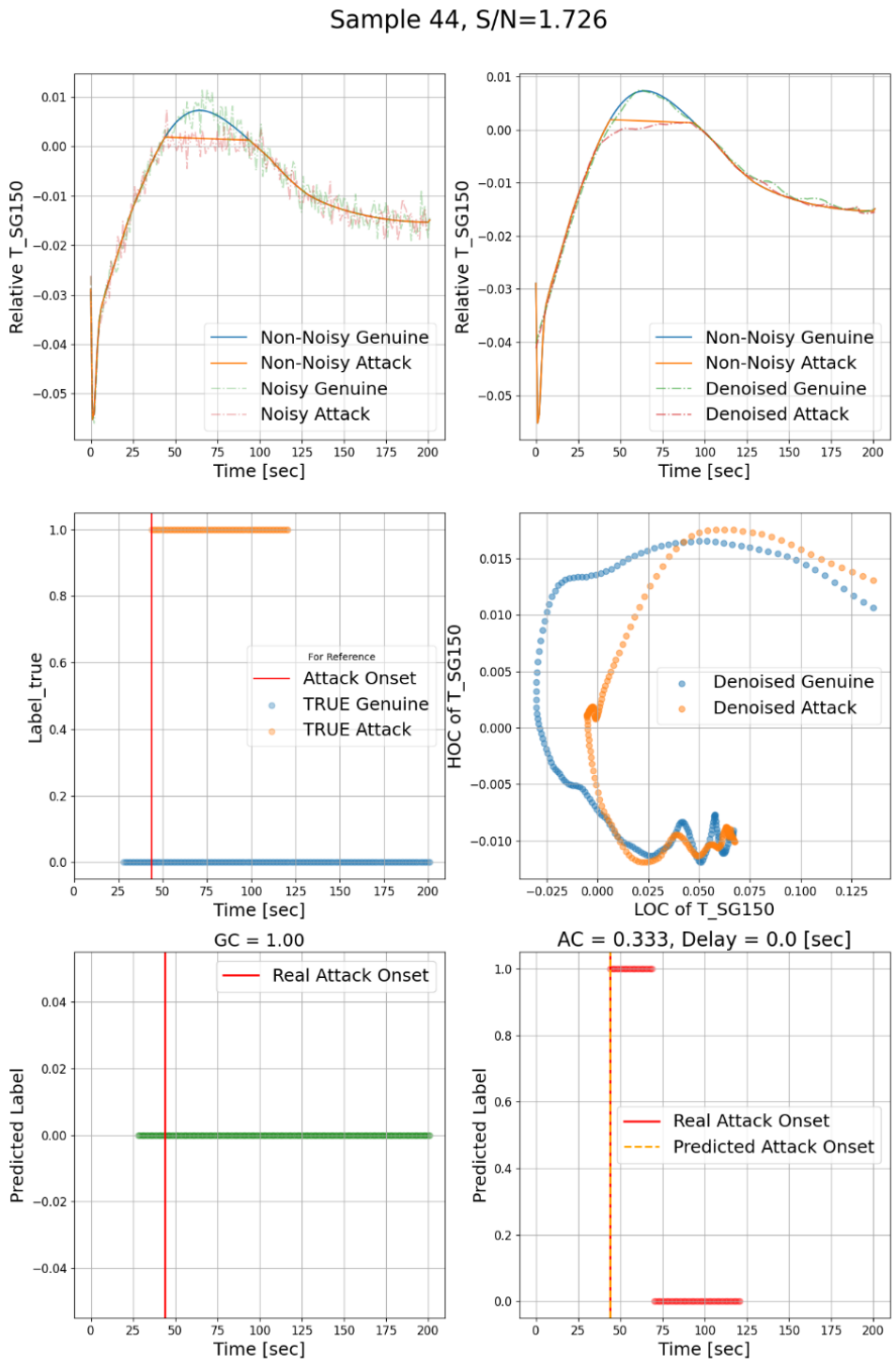

Figure 7.8 Univariate FDI Detection with Multilevel denoising (Region 2)

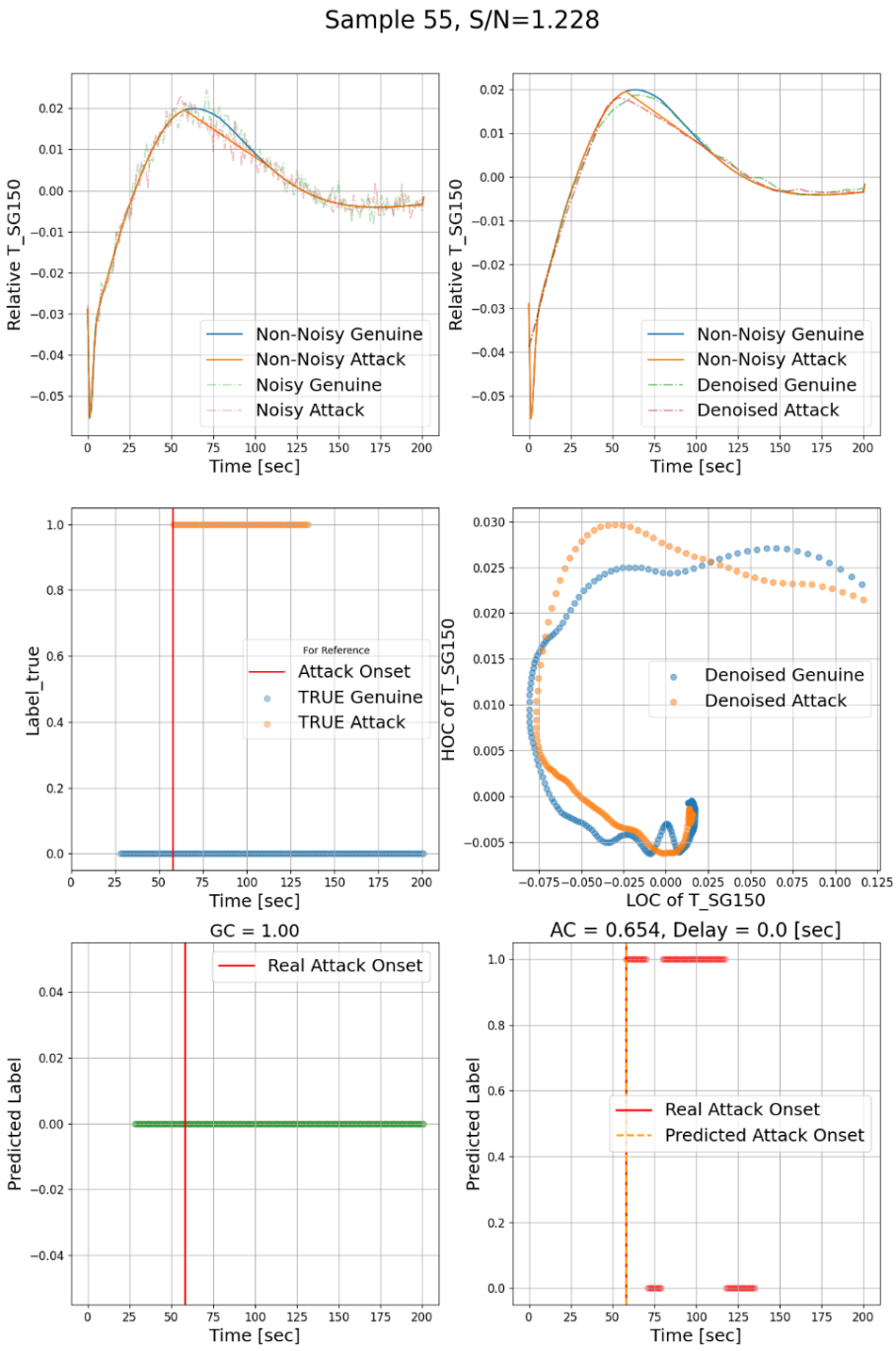

Figure 7.9 Univariate FDI Detection with Multilevel denoising (Region 3)

To have an overall assessment of the detection results for all analyzed 250 test cases, the relationship between the clean signal-to-noise ratio, S_{clean}/N, and the detection delay time is plotted in Figure 7.10. Results show a trend that for higher signal-to-noise ratio, the detection delay time will be shorter, and vice versa. The detectable limit of S_{clean}/N ratio is 0.933, shown as a red horizontal dashed line, demonstrating that when the S_{clean}/N ratio is above this limit, the attack can always be detected. For the cases that are not detected, their meaningful clean signal is very small, because the attack and the genuine track almost match each other, an example is shown in Figure 7.11.

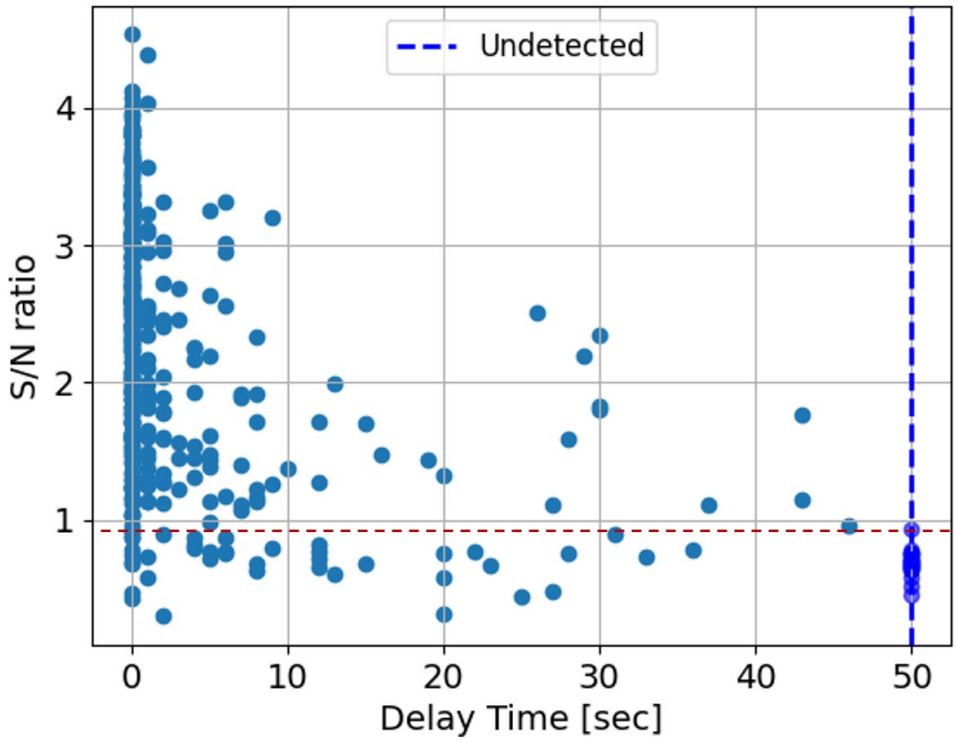

Figure 7.10 Detection Delay Time vs. Signal-to-noise Ratio with Univariate Monitoring

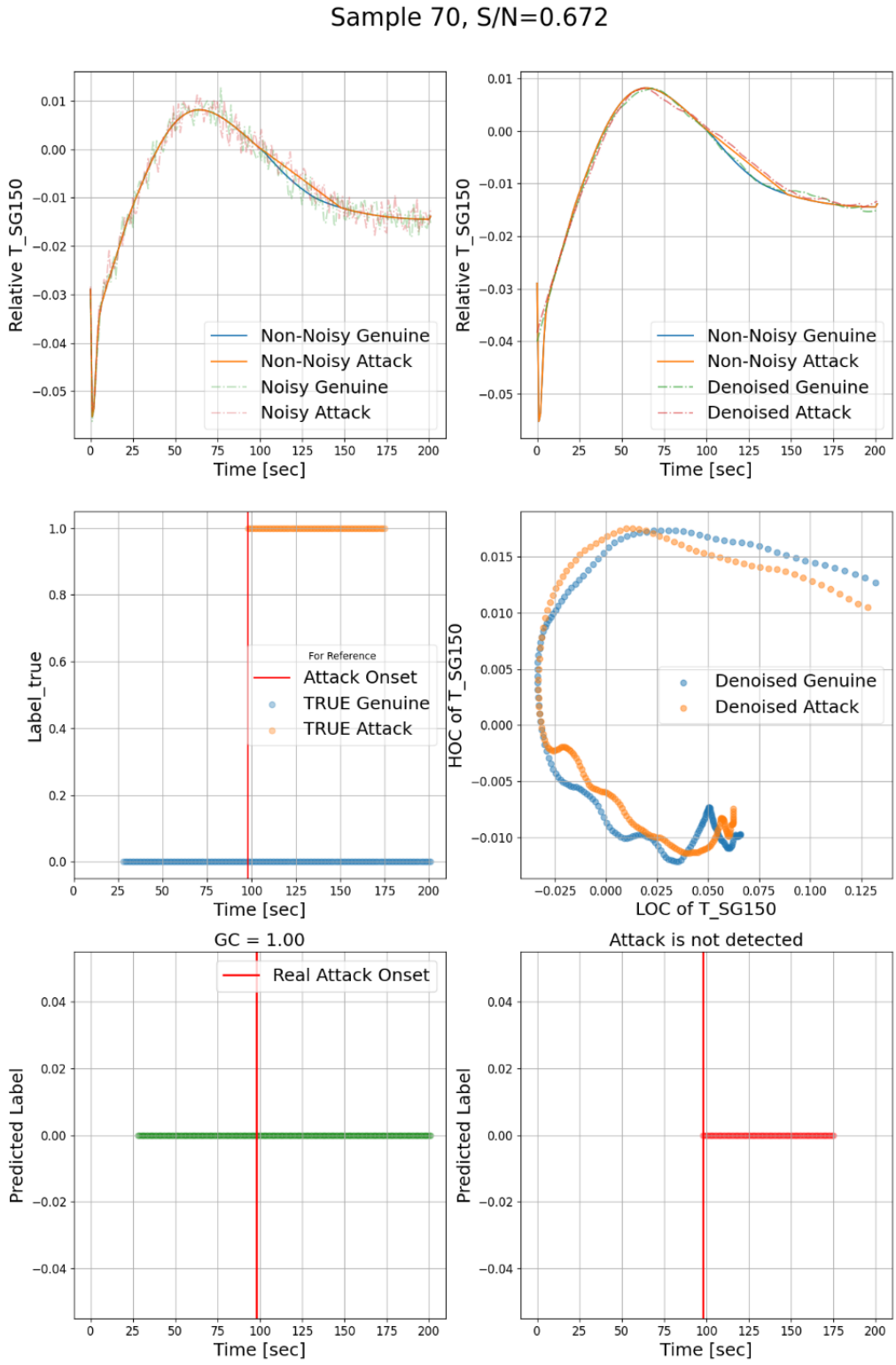

Figure 7.11 Example of Undetected Attack

7.2.4 FDI Detection with Multivariate Monitoring

The last section focused on analyzing behavior using the LOC and HOC associated with a single response, i.e., no correlation between responses is employed. In this section, the LOCs and HOCs obtained from different responses are employed to design the classifier. Intuitively, this is sought to improve the performance of the classifier. An important step here is to devise a procedure by which the best LOCs and HOCs from a group of responses are selected to train the classifier. By way of an example, consider two responses T_SG150 and T_SG210, and consider a single LOC and a single HOC associated with each one of them. A brute force approach could be used to incorporate all HOCs and LOCs from multiple responses, this is however is expected to overwhelm the training. Instead, we employ a simple criterion, that's to select the pair of features with the highest mutual information. This can be calculated using kernel density estimation of the joint probability distribution. An example is shown in Figure 7.12, where the bottom-left subplot shows the best pair of features. Here this can be eye-balled by identifying the two features showing the most correlation, i.e., least uncertainty. More sophisticated selection criterion, based on fusion of multiple features from multiple responses will be sought in our future work.

The subplots in Figure 7.12 plot the components from two process variables among all samples, in which the red area indicates where the components data accumulates, and the purple area indicates less data accumulation. T_SG150 represents the temperature at the primary side of the steam generator, while T_SG210 represents the temperature at the secondary side of the steam generator. The LOC and HOC from each process variable are denoted as $\boldsymbol{\alpha}_L$ and $\boldsymbol{\alpha}_H$ respectively. From Figure 7.13, one can see there is a trend in every subplot. In other words, the components recorded by the x-axis is dependent on the ones for the y-axis. This section focuses on a question that whether this dependence can be leveraged to find subtle inconsistency within the data. To validate this point, the same attack window in subsection 7.2.3 is injected into the temporal signal profile of T_SG150, but the feature construction involves correlated LOC and HOC from different process variables.

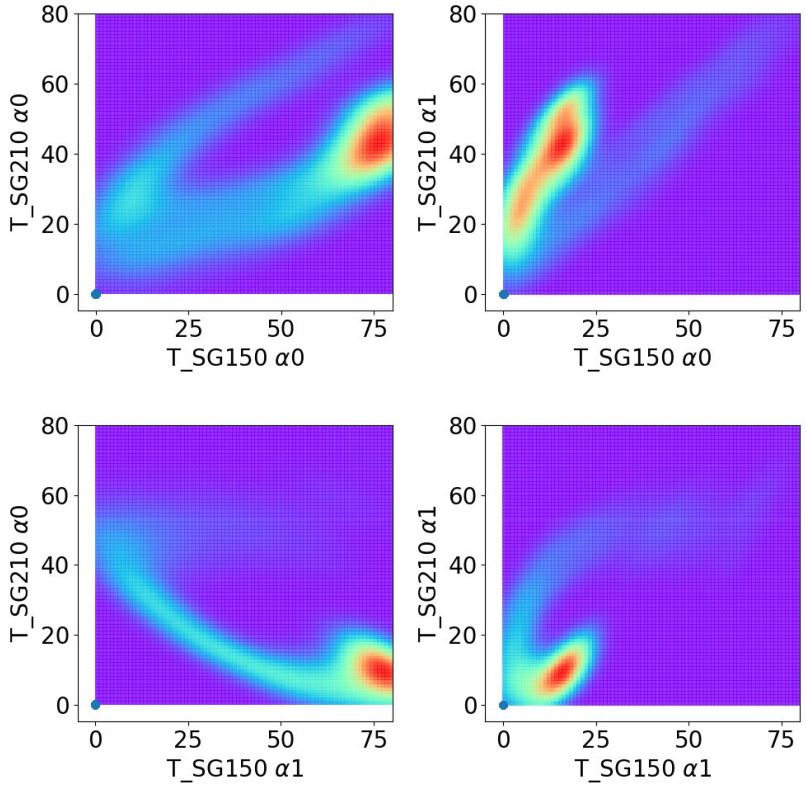

Figure 7.12 Components correlation of Different process variables

The next set of figures, Figure 7.13 to Figure 7.15 repeat the detection results but now using the LOCs and HOCs from two different responses. It is assumed that the first response T_SG150 is attacked, whereas the other response T_SG210 is not attacked. The classifier is trained based on a single LOC of the T_SG150 and a single HOC of the T_SG210 as plotted in the middle right subplot in Figure 7.13. The second subplot in Figure 7.13shows the noisy, non-noisy and denoised temporal profiles of T_SG210. The non-noisy temporal evolution of T_SG210 is shown as the orange solid curve; the green dashed line represents the T_SG210 temporal profile with evolution; the denoised profile is represented as a blue curve. From the results, one can tell that compared to univariate monitoring, the attack data detection coverage increases from AC=0.756 to AC=1.0 with two responses used for monitoring. Similar behavior is observed for the peak region, where AC increased from 0.33 to 0.821. For the decreasing region, the AC did not change much because the attack has a very similar trend to the genuine profile.

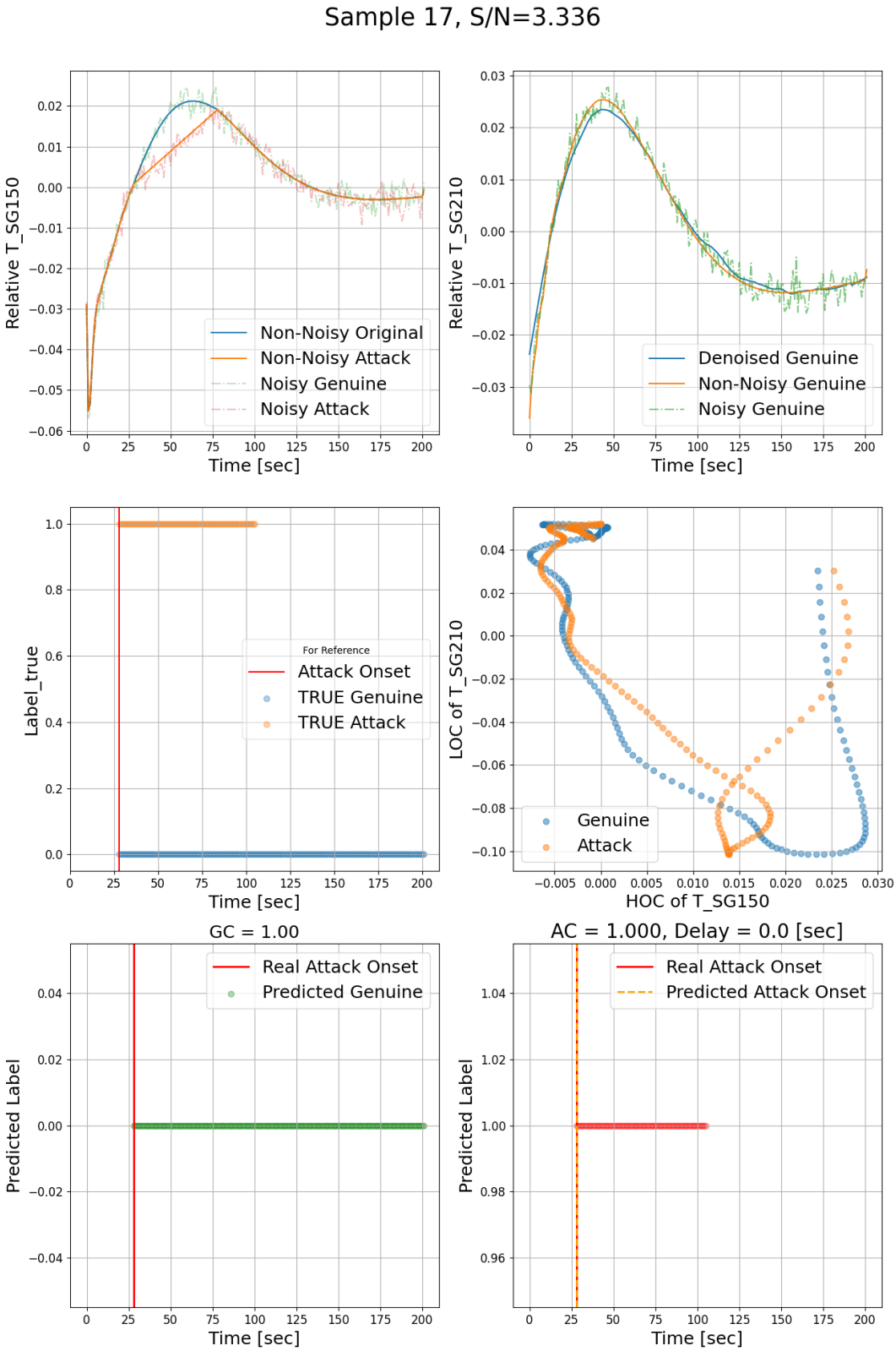

Figure 7.13 Multivariate FDI detection with Multilevel denoising (Region 1)

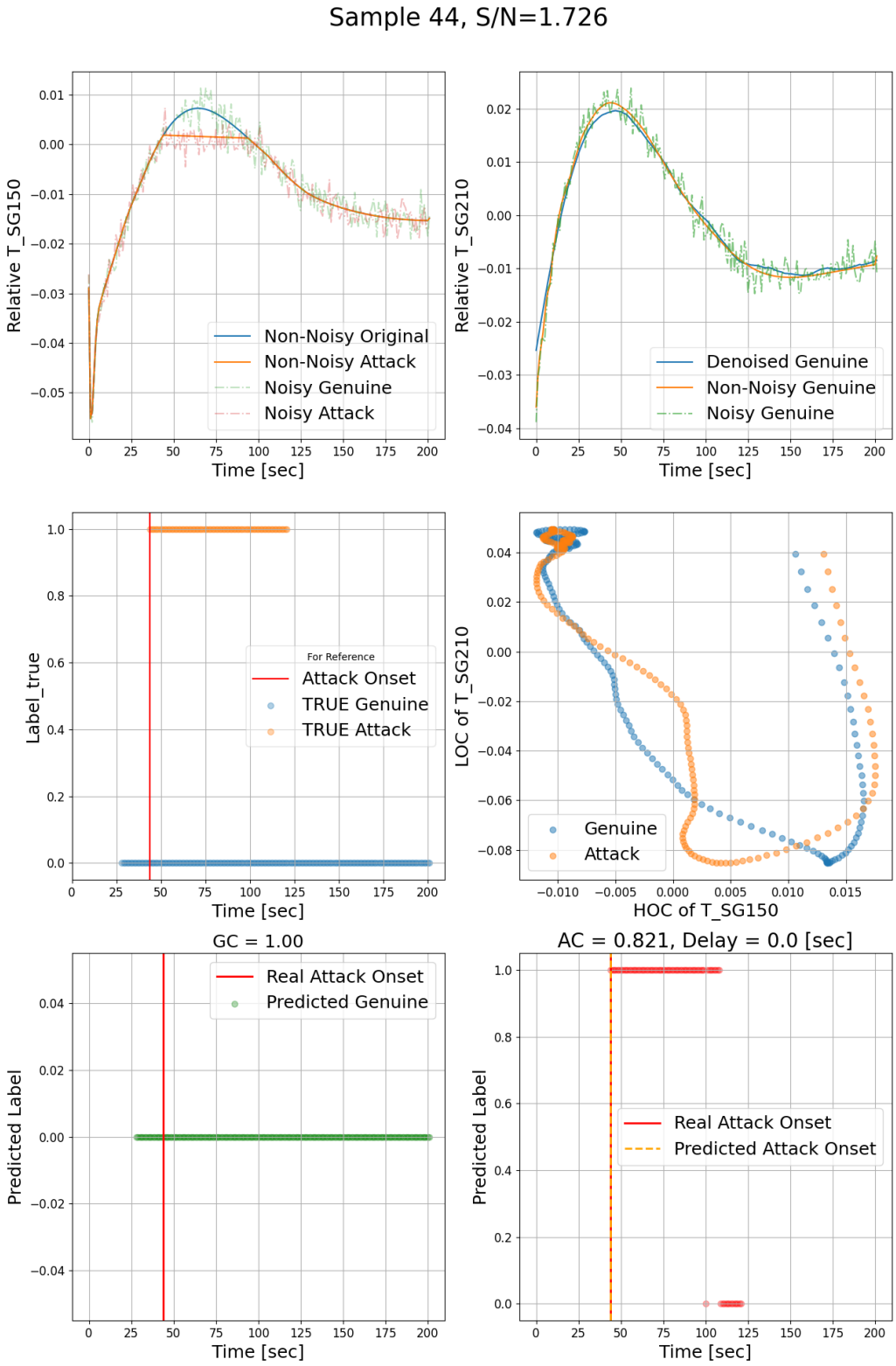

Figure 7.14 Multivariate FDI detection with Multilevel denoising (Region 2)

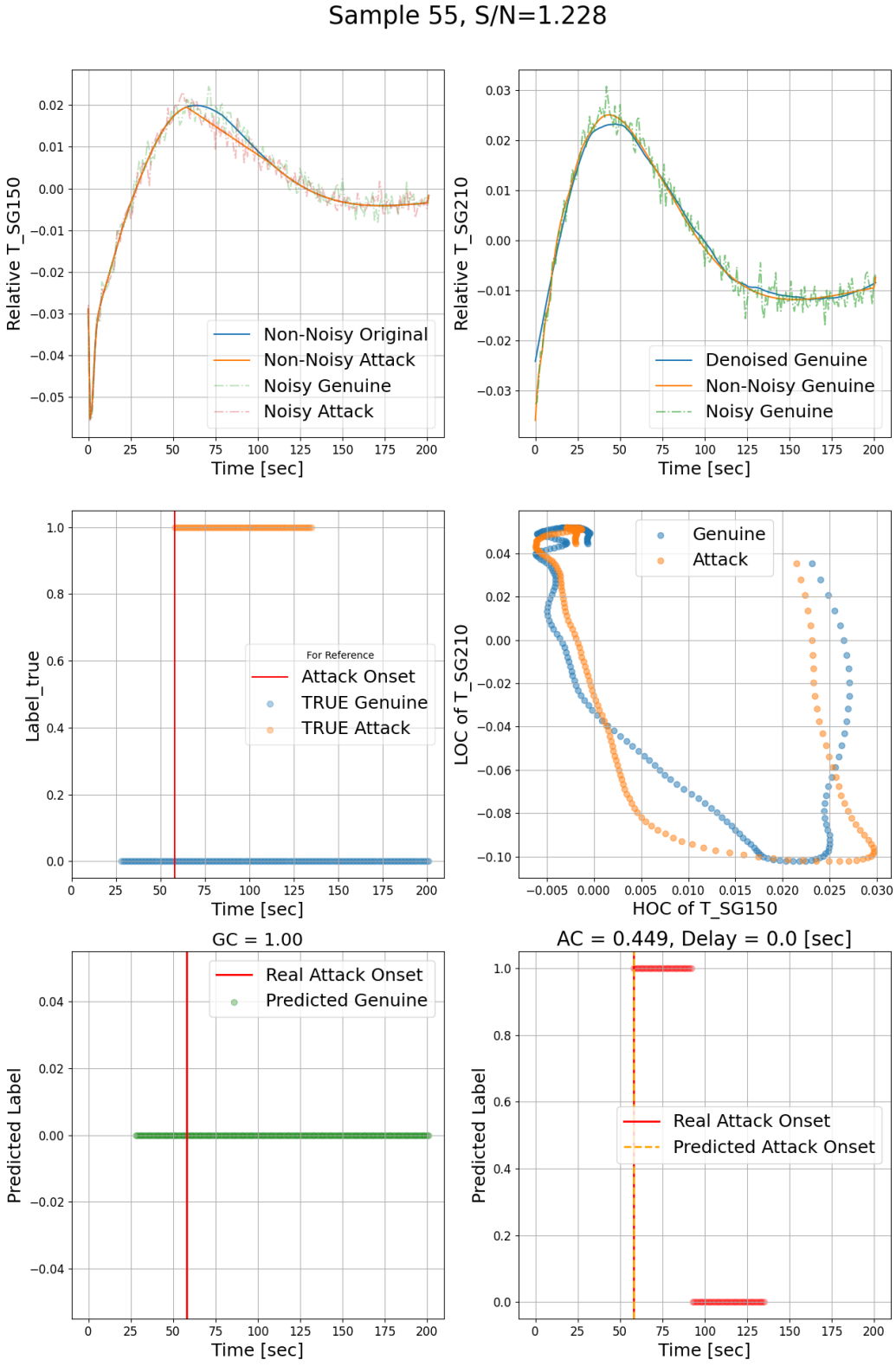

Figure 7.15 Multivariate FDI detection with Multilevel denoising (Region 3)

As the ultimate goal here is to detect as many falsified data points as possible and to issue the alarm as early as possible. The multivariate monitoring does not always lead to a superior detection result in every aspect than univariate monitoring. To evaluate the overall performance of the multivariate, Figure 7.16 represents the relationship between delayed detection time and the signal-to-noise ratio. For multivariate monitoring, the S_{clean}/N detectable limit is 0.726, shown as the red dashed line, which is lower than the limit of univariate monitoring. Figure 7.17 shows the histogram of detection delay of both monitoring results, which indicates the univariate monitoring has a small superiority on detection delay. Scatter plots in Figure 7.18 represent the relationship between two *S/N* ratios, in which different colors indicate the different time periods of detection delay. Most attack vectors can be detected within 10 seconds, shown as yellow dots and there are less long-time detection delay cases in multivariate monitoring. And for both monitoring approaches, the cases with a longer detection time mostly locate at a low S/N region. From the perspective of AC and GC, shown in Figure 7.19, one can tell that the multivariate monitoring can lead to a longer warning time.

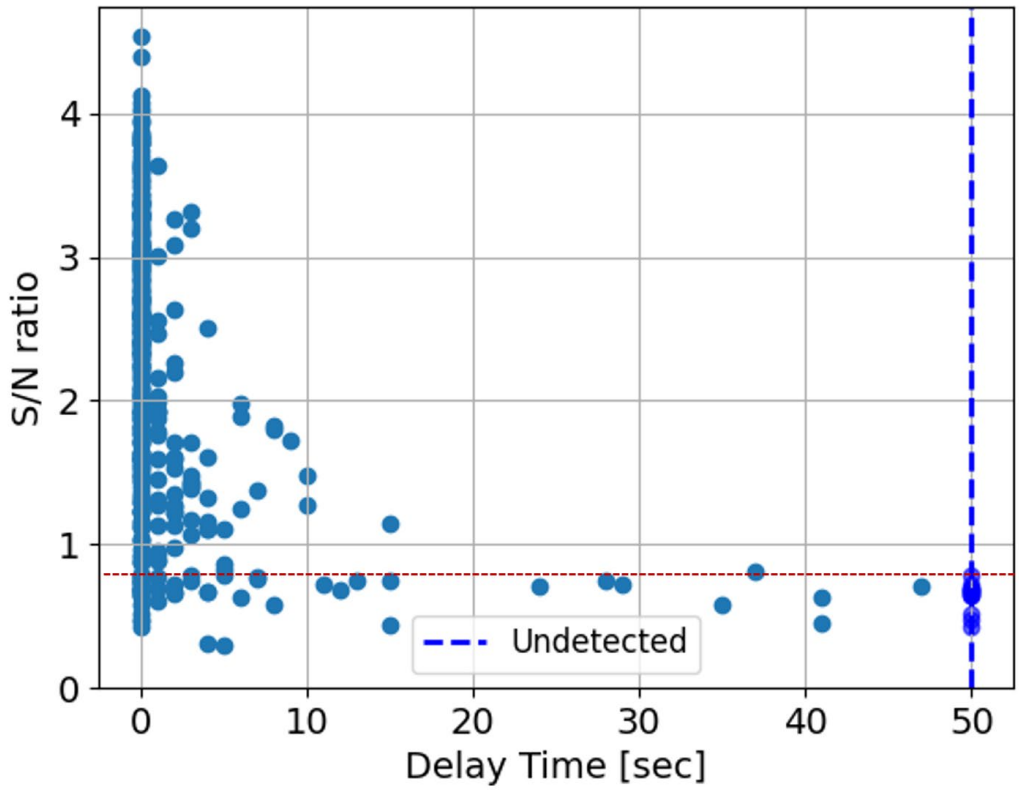

Figure 7.16 Detection Delay Time vs. S/N Ratio with Multilevel denoising

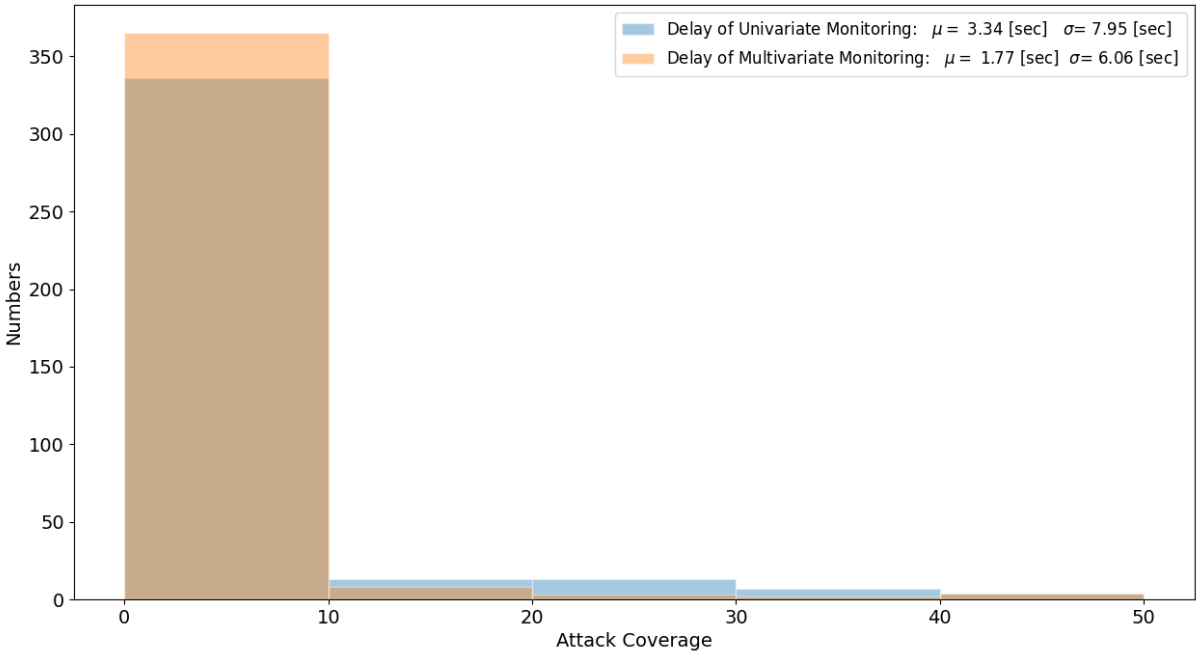

Figure 7.17 Histogram Comparison of Detection Delay

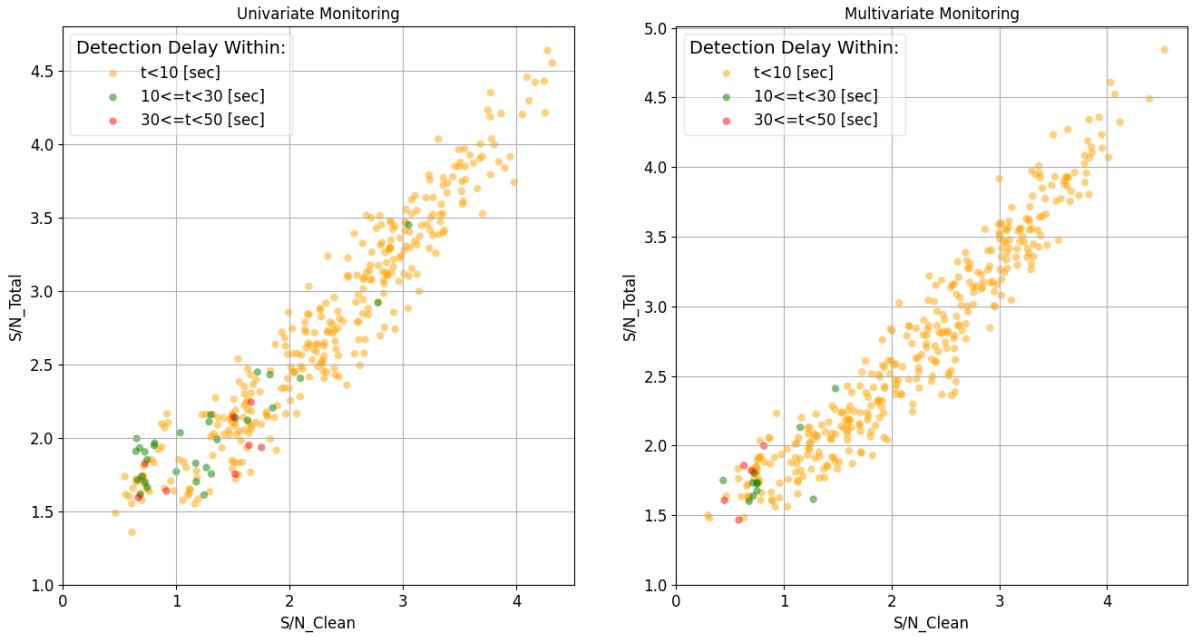

Figure 7.18 Relationship between Clean and Total S/N ratio with Detection Delay

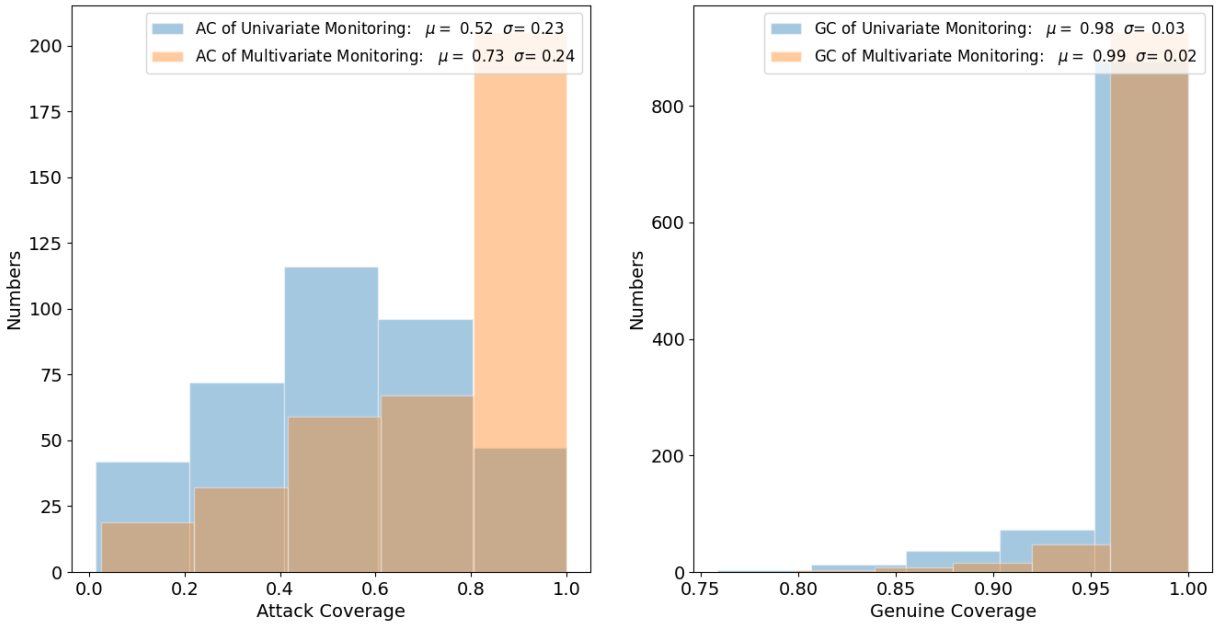

Figure 7.19 Histogram of AC and GC for both monitoring methods

7.3 Results Summary

In this chapter, a comparison study follows a denoising result analysis to investigate the detection for subtle triangle FDI attack vector. The results indicate the effectiveness of the detection algorithm for most simulated attack vectors. Moreover, the multilevel denoising approach can shorten the detection delay time and identify more imperceptible attack vectors as shown in Figure 7.8 and Figure 7.9. From the macroscopic perspective, the multilevel denoising method excels the single level one due to more identified attacks vectors and shorter detection delay. In 7.2.4, the detection results with one more process variable is demonstrated. The feature vector contains both of LOC and HOC from two correlated responses. The results indicate that engaging extra information will help the classifier distinguish the falsified signals from the genuine ones. Before concluding, it is important to remark that the computational cost for the proposed detection algorithm may be split into two components, an offline and an online component. The off-line component comprises the computational cost to design the HOCs and LOCs which can be done during a training phase. Further, because all the attack scenarios are based on the idea of triangle attacks, the training of the classifier can also be completed offline. This represents the key cost of the overall algorithm as the online component involves only the projection of the windowed time-series over the HOC/LOC components (which are simple inner product operations) and the execution of the already-trained classifier, both are essentially instantaneous.

8 CONCLUSION

Industrial control systems are currently being upgraded with digital instrumentations for efficient control, operational convenience, and expeditious data traffic. Despite the numerous benefits of digitization, one must address the threats posed by potential adversaries looking for vulnerabilities to exploit. The preliminary study demonstrates two key results: (1) the attacker can learn the system behavior to a first approximation solely with historical data; (2) equipped with an approximate physics model and historical data, the attacker can recover the missing details of the model, e.g. model parameters and make accurate predictions of the system behavior. To address this threat, the exploratory study presents an OT defense developing unique signatures for the individual systems and calculated using machine learning techniques as guided by high-fidelity physics model and the system-specific design and historical operational data. With learned system behavior, attackers can launch stealthy attacks to circumvent the detection by conventional monitoring techniques. The exploratory study provides a basic detection algorithm based on single and multiple-responses, demonstrating how subtle variations can be detected by analyzing both the HOCs and LOCs. In practice the number of sensors in a nuclear power plant is very high, implying that a brute force application of the proposed approach for each sensor will not be computationally feasible. Thus, in practice, another algorithm must be developed to select HOCs and LOCs across many sensors. To achieve that, standard SVD-based decomposition techniques can be applied on multi-response data, where the idea here is not to reduce the dimensionality but to capture a number of LOCs and HOCs that take into account the correlations across the sensors data. A window-based approach, akin to the RWD algorithm employed here, will be used to generate a joint PDF of candidate pairs of LOCs and HOCs. Next, information-theoretic metrics, such as mutual information, will be employed to identify the candidate pairs with the highest mutual information. The idea is to rely on both dominant, referred to as the LOCs, as well as less dominant features, referred to as the HOCs, to derive signatures that can identify FDI attacks. Specifically, this idea is implemented as an offline analysis approach in the first place, and consequently improved as a robust online monitoring toolkit in conjunction with a novel denoising technique. Results indicate that the patterns established by the LOCs and HOCs are effective in detecting subtle variations, expected to be the mode of attack during an initial lie-in-wait period, used to test the attacker's ability to bypass detection. Results also indicate that attacks comparable

to the noise level can be detected, with the detection ability improved with additional responses used for monitoring. This is especially important for replay attack, which rely on using older genuine data to spoof future sensors readings. Several outstanding developments need to be further addressed in support of this work. For example, a more detailed analysis of the binary classification results is needed to design a better detection criterion by analyzing the relationship of the points trigger the alarm to the classifier's decision boundary.

9 FUTURE WORK

Aforementioned limitations of this work are mainly on two aspects. The first one is that this monitoring algorithm can only identify the attacks/anomaly that deviate from the patterns of the genuine signals. One way to solve this issue is introducing active monitoring techniques, e.g., physical watermark etc., whose effectiveness have been proved. Another way is to develop online adaptive pattern discovery and fault/attack learning capabilities. The other limitation is the detection effectiveness for attack vectors in the null space of identified LOCs and HOCs. Future work will expand this work to the defense for this type of attacks.

Besides, for online monitoring toolkit, one has to construct feature vector after signal denoising is accomplished, which would add more delay time for the attack detection. Future work will include the minimize the delay time resulting from denoising. In this study, the way of employing the LOCs and HOCs components across multiple responses is naïve. Future work will also optimize the integration of HOCs components across multiple process variables to maximize the classification accuracy.

This work focuses on the subtle triangle attack vector injected in a single process variable, while the attacker with intimate knowledge can launch coordinated attacks for multiple response. For long-term goal, future work will expand the attack scenarios and develop the current algorithm into a framework to accommodate the needs of online monitoring such as early fault/degradation diagnosis. The algorithm implementation to detect component degradation is demonstrated in the Appendix.

REFERENCE

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

[2] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid Defense Use Case," *Electr. Inf. Shar. Anal. Cent.*, p. 36, 2016, [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf.

[3] K. E. Hemsley and D. R. E. Fisher, "History of Industrial Control System Cyber Incidents," *INL/CON-18-44411-Revision-2*, no. December, pp. 1–37, 2018, [Online]. Available: https://www.osti.gov/servlets/purl/1505628.

[4] NIST, "Improving Critical Infrastructure Cybersecurity Executive Order 13636 Preliminary Cybersecurity Framework."

[5] A. Fawzy and H. M. O. Mokhtar, "Outliers detection and classification in wireless sensor networks," *Egypt. Informatics J.*, vol. 14, no. 2, pp. 157–164, 2013, doi: 10.1016/j.eij.2013.06.001.

[6] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier," *Neurocomputing*, vol. 150, pp. 289–303, 2015, doi: https://doi.org/10.1016/j.neucom.2014.05.086.

[7] F. Smarra, A. Jain, R. Mangharam, and A. D'Innocenzo, "Data-driven Switched Affine Modeling for Model Predictive Control," *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 199–204, 2018, doi: 10.1016/j.ifacol.2018.08.034.

[8] D. Hadžiosmanovi, R. Sommer, and P. H. Hartel, "Through the Eye of the PLC : Semantic Security Monitoring for Industrial Processes."

[9] X. Niu, J. Li, and J. Sun, "Dynamic Detection of False Data Injection Attack in Smart Grid using Deep Learning."

[10] E. Trunzer *et al.*, "Failure Mode Classification for Control Valves for Supporting Data-Driven Fault Detection," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2017-Decem, pp. 2346–2350, 2018, doi: 10.1109/IEEM.2017.8290311.

[11] A. Sundaram, H. S. Abdel-Khalik, and Oussama Ashy, "Exploratory Study into the Effectiveness of Active Monitoring Techniques."

[12] D. I. Urbina *et al.*, "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, 2016, no. c, pp. 1092–1105, doi: 10.1145/2976749.2978388.

[13] M. Grieves, “Digital Twin : Manufacturing Excellence through Virtual Factory Replication A Whitepaper by Dr . Michael Grieves,” 2014, doi: 10.1016/B978-0-12-382196-6.00017-0.

[14] Juan Lopez Jr, “Digital Twin Framework for Power Grid Cyber Resilience,” 2018, Accessed: Nov. 12, 2018. [Online]. Available: https://agenda.icscybersecurityconference.com/event/GcFU/digital-twin-framework-for-power-grid-cyber-resilience.

[15] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, 2011, doi: 10.1145/1952982.1952995.

[16] “Passive monitoring.” [Online]. Available: https://en.wikipedia.org/wiki/Passive_monitoring.

[17] “Synthetic monitoring.” https://en.wikipedia.org/wiki/Synthetic_monitoring.

[18] Shannon L. Eggers, “Adapting Anomaly Detection Techniques for Online Intrusion Detection in Nuclear Facilities,” 2018.

[19] F. Zhang, J. W. Hines, and J. B. Coble, “A Robust Cybersecurity Solution Platform Architecture for Digital Instrumentation and Control Systems in Nuclear Power Facilities,” *Nucl. Technol.*, vol. 206, no. 7, pp. 939–950, 2020, doi: 10.1080/00295450.2019.1666599.

[20] W. Wang, F. Di Maio, and E. Zio, *A non-parametric cumulative sum approach for online diagnostics of cyber attacks to nuclear power plants*. 2019.

[21] H. L. Gawand, A. K. Bhattacharjee, and K. Roy, “Securing a Cyber Physical System in Nuclear Power Plants Using Least Square Approximation and Computational Geometric Approach,” *Nucl. Eng. Technol.*, vol. 49, no. 3, pp. 484–494, 2017, doi: 10.1016/j.net.2016.10.009.

[22] Y. Zhao, L. Huang, C. Smidts, and Q. Zhu, “A game theoretic approach for responding to cyber-attacks on nuclear power plants,” *11th Nucl. Plant Instrumentation, Control. Human-Machine Interface Technol. NPIC HMIT 2019*, pp. 399–410, 2019.

[23] P. K. Vaddi, Y. Zhao, X. Diao, and C. Smidts, “Dynamic Bayesian networks based event-classifier in support for reactor operators in case of cyber-security threats,” *11th Nucl. Plant Instrumentation, Control. Human-Machine Interface Technol. NPIC HMIT 2019*, pp. 1380–1395, 2019.

[24] R. S. Smith, “Covert Misappropriation Control Systems,” *Control Syst. 2015 Issue 1*, no. January 2015, pp. 82–92, 2015.

[25] H. Sandberg and K. H. Johansson, “On Security Indices for State Estimators in Power Networks.”

[26] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, “Cyber security analysis of state estimators in electric power systems,” *Proc. IEEE Conf. Decis. Control*, pp. 5991–5998, 2010, doi: 10.1109/CDC.2010.5717318.

[27] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on the smart grid,” *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011, doi: 10.1109/TSG.2011.2163807.

[28] O. Vukovi, “On the Security of Distributed Power System State Estimation under Targeted Attacks,” pp. 666–672, 2013.

[29] J. A. Quinn and M. Sugiyama, “A least-squares approach to anomaly detection in static and sequential data,” *Pattern Recognit. Lett.*, vol. 40, no. 1, pp. 36–40, 2014, doi: 10.1016/j.patrec.2013.12.016.

[30] M. Krotofil and A. A. Cárdenas, “Resilience of process control systems to cyber-physical attacks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8208 LNCS, pp. 166–182, 2013, doi: 10.1007/978-3-642-41488-6_12.

[31] G. Dan and H. Sandberg, “Stealth Attacks and Protection Schemes for State Estimators in Power Systems,” pp. 214–219, 2010, doi: 10.1109/SMARTGRID.2010.5622046.

[32] A. Giani, E. Bitar, M. Garcia, and M. Mcqueen, “Smart Grid Data Integrity Attacks : Characterizations and Countermeasures π,” *2011 IEEE Int. Conf. Smart Grid Commun.*, no. 025478, pp. 232–237, 2011, doi: 10.1109/SmartGridComm.2011.6102324.

[33] T. T. Kim and H. V. Poor, “Strategic Protection Against Data Injection Attacks on Power Grids,” *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011, doi: 10.1109/TSG.2011.2119336.

[34] J. Li and X. Huang, “Cyber attack detection of I&C systems in NPPS based on physical process data,” *Int. Conf. Nucl. Eng. Proceedings, ICONE*, vol. 2, pp. 1–4, 2016, doi: 10.1115/ICONE24-60773.

[35] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, “SRID: State relation based intrusion detection for false data injection attacks in SCADA,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8713 LNCS, no. PART 2, pp. 401–418, 2014, doi: 10.1007/978-3-319-11212-1_23.

[36] S. Mclaughlin, “CPS : Stateful Policy Enforcement for Control System Device Usage,” pp. 109–118, 2015.

[37] D. D. Yao, “Orpheus : Enforcing Cyber-Physical Execution Semantics to Defend Against Data-Oriented Attacks,” pp. 315–326, 2017.

[38] 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,” *Prog. Nucl. Energy*, vol. 124, no. December 2019, p. 103339, 2020, doi: 10.1016/j.pnucene.2020.103339.

[39] Y. Zhao and C. Smidts, “A control-theoretic approach to detecting and distinguishing replay attacks from other anomalies in nuclear power plants,” *Prog. Nucl. Energy*, vol. 123, no. December 2019, p. 103315, 2020, doi: 10.1016/j.pnucene.2020.103315.

[40] F. Zhang and J. B. Coble, “Robust localized cyber-attack detection for key equipment in nuclear power plants,” *Prog. Nucl. Energy*, vol. 128, no. July, p. 103446, 2020, doi: 10.1016/j.pnucene.2020.103446.

[41] Y. Li, H. Abdel-Khalik, A. J. Brunett, E. Jennings, T. Mui, and R. Hu, “ROM-based Surrogate Systems Modeling of EBR-II,” *Nucl. Sci. Eng.*, 2020, doi: 10.1080/00295639.2020.1840238.

[42] Y. Li and H. S. Abdel-Khalik, “Data trustworthiness signatures for nuclear reactor dynamics simulation,” *Prog. Nucl. Energy*, vol. 133, 2021.

[43] J. Larsen, “Miniaturization,” 2014.

[44] Y. Li, E. Bertino, and H. Abdel-Khalik, “Effectiveness of Model-Based Defenses for Digitally Controlled Industrial Systems : Nuclear Reactor Case Study,” *Nucl. Technol.*, vol. 206, no. 1, pp. 82–93, 2018.

[45] R. Rai and C. K. Sahu, “Driven by Data or Derived through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques with Cyber-Physical System (CPS) Focus,” *IEEE Access*, vol. 8, pp. 71050–71073, 2020, doi: 10.1109/ACCESS.2020.2987324.

[46] “Computer Sabotage at Nuclear Power Plant.” https://www.risidata.com/Database/Detail/computer_sabotage_at_nuclear_power_plant#:~:text=A computer programmer at the,a charge of premeditated sabotage.

[47] “Slammer Impact on Ohio Nuclear Plant,” [Online]. Available: https://www.risidata.com/Database/Detail/slammer-impact-on-ohio-nuclear-plant.

[48] “Texas Power Company Hack.” https://www.risidata.com/Database/Detail/texas_power_company_hack.

[49] R. M. Lee, M. J. Assante, and T. Conway, “ICS CP/PE (Cyber-to-Physical or Process Effects) case study paper – German Steel Mill Cyber Attack,” *SANS, Ind. Control Syst.*, p. 15, 2014.

[50] “Monju power plant facility PC infected with virus,” *Japan Today*. https://japantoday.com/category/national/monju-power-plant-facility-pc-infected-with-virus.

[51] J. Park and M. Cho, “South Korea blames North Korea for December hack on nuclear operator,” *Reuters*. https://www.reuters.com/article/us-nuclear-southkorea-northkorea/south-korea-blames-north-korea-for-december-hack-on-nuclear-operator-idUSKBN0MD0GR20150317.

[52] J. Min, “North Korea’s Asymmetric Attack on South Korea’s Nuclear Power Plants.” http://large.stanford.edu/courses/2017/ph241/min1/.

[53] B. Matthew, “Scenarios of Insider Threats To Japan’S Nuclear Facilities and Materials – and Steps To Strengthen Protection S Nuclear Facilities and Materials – and Steps To Strengthen Protection Materials – and Steps To Strengthen Protec,” pp. 1–19, 2017.

[54] C. Steitz and E. Auchard, “German nuclear plant infected with computer viruses, operator says,” 2016. https://www.reuters.com/article/us-nuclearpower-cyber-germany/german-nuclear-plant-infected-with-computer-viruses-operator-says-idUSKCN0XN2OS.

[55] “Former U.S. Nuclear Regulatory Commission Employee Pleads Guilty to Attempted Spear-Phishing Cyber-Attack on Department of Energy Computers.” https://www.justice.gov/opa/pr/former-us-nuclear-regulatory-commission-employee-pleads-guilty-attempted-spear-phishing-cyber.

[56] “2017 cyberattacks on Ukraine.” https://en.wikipedia.org/wiki/2017_cyberattacks_on_Ukraine.

[57] J. Fruhlinger, “Petya ransomware and NotPetya malware: What you need to know now.” https://www.csoonline.com/article/3233210/petya-ransomware-and-notpetya-malware-what-you-need-to-know-now.html.

[58] D. Das, “An Indian nuclear power plant suffered a cyberattack. Here’s what you need to know.,” *the Washington Post*. https://www.washingtonpost.com/politics/2019/11/04/an-indian-nuclear-power-plant-suffered-cyberattack-heres-what-you-need-know/.

[59] “References for Cyber Incidents at Nuclear Facilities.” .

[60] “RISI Online Incident Database,” [Online]. Available: https://www.risidata.com/Database.

[61] “Significant Cyber Events List.” [Online]. Available: https://csis-website-prod.s3.amazonaws.com/s3fs-public/201106_Significant_Cyber_Events_List.pdf.

[62] T. MacLean, R. Borrelli, and M. Haney, “Cyber Security Modeling of Non-Critical Nuclear Power Plant Digital Instrumentation,” pp. 87–100, 2019, doi: 10.1007/978-3-030-34647-8_5.

[63] J. G. Song, J. W. Lee, G. Y. Park, K. C. Kwon, D. Y. Lee, and C. K. Lee, “An analysis of technical security control requirements for digital I&C systems in nuclear power plants,” *Nucl. Eng. Technol.*, vol. 45, no. 5, pp. 637–652, 2013, doi: 10.5516/NET.04.2012.091.

[64] C. K. Lee, *Introduction of a cyber security risk analysis and assessment system for digital I&C systems in nuclear power plants*, vol. 46, no. 9. IFAC, 2013.

[65] B. Kesler, "The Vulnerability of Nuclear Facilities to Cyber Attack," *Strateg. Insights*, vol. 10, no. 1, pp. 15–25, 2011, [Online]. Available: http://large.stanford.edu/courses/2015/ph241/holloway1/docs/SI-v10-I1_Kesler.pdf.

[66] A. Varuttamaseni, R. A. Bari, and R. Youngblood, "Construction of a Cyber Attack Model for Nuclear Power Plants 10th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies," 2017.

[67] K. Sturzebecher, "DRAFT REGULATORY GUIDE DG-1206, 'CONFIGURATION MANAGEMENT PLANS FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS,'" 2012.

[68] Karl Sturzebecher and T. H. Boyce, "Software Requirement Specifications for Digital Computer Software and Complex Electronics Used in Safety Systems of Nuclear Power Plants," *Federal Register*, vol. 77, no. 163, p. 50726, 2012.

[69] C. S. Glantz *et al.*, *NUREG/CR-6847, "Cyber Security Self-Assessment Method for U.S. Nuclear Power Plants."* 2004.

[70] NRC, "Protection of digital computer and communication systems and networks," *10 CFR 73.54*, 2009. https://www.nrc.gov/reading-rm/doc-collections/cfr/part073/part073-0054.html.

[71] NEI, "NEI 08-09, 'Cyber Security Plan for Nuclear Power Reactors,'" no. April, 2010.

[72] NRC, "Cyber security event notifications," *10 CFR 73.77*, 2015. https://www.nrc.gov/reading-rm/doc-collections/cfr/part073/part073-0077.html.

[73] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014, doi: 10.1016/j.sigpro.2013.12.026.

[74] V. Chandola, A. BANERJEE, and V. KUMAR, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–72, 2009, [Online]. Available: http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf.

[75] F. V. Jensen, "Bayesian networks," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 1, no. 3, pp. 307–315, 2009, doi: 10.1002/wics.48.

[76] J. R. Bence, "Analysis of short time series: Correcting for autocorrelation," *Ecology*, vol. 76, no. 2, pp. 628–639, 1995, doi: 10.2307/1941218.

[77] T. K. Ho, "Random decision forests," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 278–282, 1995, doi: 10.1109/ICDAR.1995.598994.

[78] P. A. A. Resende and A. C. Drummond, “A survey of random forest based methods for intrusion detection systems,” *ACM Comput. Surv.*, vol. 51, no. 3, 2018, doi: 10.1145/3178582.

[79] C. Cortes and V. Vapnik, “A fast SVD for multilevel block Hankel matrices with minimal memory storage,” *Numer. Algorithms*, vol. 69, pp. 875–891, 2015, [Online]. Available: https://doi.org/10.1007/s11075-014-9930-0.

[80] A. J. SMOLA and B. SCHOLKOPF, “A tutorial on support vector regression,” *Statistics and Computing*, 2004. .

[81] J.-P. Vert, K. Tsuda, and B. Scholkopf, “A Primer on Kernel Methods,” in *Kernel Methods in Computational Biology*, MIT Press, 2004, pp. 35–70.

[82] R. M. Neal, “Regression and Classification Using Gaussian Process Priors,” *Bayesian Stat.*, vol. 6, pp. 475–501, 1998.

[83] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2003-Janua, pp. 958–963, 2003, doi: 10.1109/ICDAR.2003.1227801.

[84] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[85] F. K. Dosilovic, M. Brcic, and N. Hlupic, “Explainable artificial intelligence: A survey,” *2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc.*, no. May, pp. 210–215, 2018, doi: 10.23919/MIPRO.2018.8400040.

[86] Y. Jia, “ML at Facebook : An Infrastructure View.”

[87] M. ‘Arif, H. Hassan, D. Nasien, and H. Haron, “A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 2, pp. 204–212, 2015, doi: 10.14569/ijacsa.2015.060230.

[88] C. C. M. Yeh *et al.*, *Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile*, vol. 32, no. 1. Springer US, 2018.

[89] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution.*, vol. 56, no. 194. 1991.

[90] Y. Meyer, “Wavelets and Operators,” *Wavelets and Operators*. 1993, doi: 10.1017/cbo9780511623820.

[91] N. E. Huang *et al.*, “The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998, doi: 10.1098/rspa.1998.0193.

[92] M. Heideman, D. Johnson, and C. S. Burrus, “Gauss and the history of the Fast Fourier Transform,” *IEEE Signal Process. Mag.*, vol. 1, no. 3, pp. 14–21, 1984.

[93] H. Abdi and L. J. Williams, “Principal component analysis. wiley interdisciplinary reviews: computational statistics,” *Wiley Interdisplinary Rev. Comput. Stat.*, pp. 1–47, 2010.

[94] K. Pearson, “ LIII. On lines and planes of closest fit to systems of points in space ,” *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901, doi: 10.1080/14786440109462720.

[95] B. C. Moore, “Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction,” *IEEE Trans. Automat. Contr.*, vol. AC-26, no. 1, pp. 17--32, 1981.

[96] T. Hofmann, B. Schölkopf, and A. J. Smola, “Kernel methods in machine learning,” *Ann. Stat.*, vol. 36, no. 3, pp. 1171–1220, 2008, doi: 10.1214/009053607000000677.

[97] B. Schölkopf, A. Smola, and K. R. Müller, “Kernel principal component analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1327, pp. 583–588, 1997, doi: 10.1007/bfb0020217.

[98] L. Blouvshtein and D. Cohen-Or, “Outlier detection for robust multi-dimensional scaling,” *arXiv*, pp. 1–10, 2018.

[99] B. Ghojogh *et al.*, “Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review,” 2019, [Online]. Available: http://arxiv.org/abs/1905.02845.

[100] E. Bair, T. Hastie, D. Paul, and R. Tibshirani, “Prediction by supervised principal components,” *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 119–137, 2006, doi: 10.1198/016214505000000628.

[101] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.

[102] M. Welling, “Fisher linear discriminant analysis,” IEEE, Feb. 2005. [Online]. Available: https://www.ics.uci.edu/~welling/teaching/273ASpring09/Fisher-LDA.pdf.

[103] A. Hyvärinen, *Independent component analysis*. New York : J. Wiley, 2001.

[104] A. Hyvärinen, “Independent component analysis : recent advances,” *Phil. Trans. R. Soc. A*, 2013.

[105] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Zolghadri Jahromi, “Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds,” *Pattern Recognit.*, vol. 44, no. 7, pp. 1357–1371, 2011, doi: 10.1016/j.patcog.2010.12.015.

[106] S. V. Stehman, “Selecting and interpreting measures of thematic Classification accuracy,” *Remote Sens. Environ.*, vol. 62, no. 1, pp. 77–89, 1997, [Online]. Available: https://doi.org/10.1016/S0034-4257(97)00083-7.

[107] Y. Li and H. Abdel-khalik, “ROM-based Subset Selection Algorithm for Efficient Surrogate Modeling,” vol. 115, pp. 1740–1743, 2016.

[108] J. A. Cook, R. C. Smith, J. M. Hite, R. Stefanescu, and J. Mattingly, “Application and evaluation of surrogate models for radiation source search,” *Algorithms*, vol. 12, no. 2, pp. 1–24, 2019, doi: 10.3390/A12120269.

[109] S. Habib, K. Heitmann, D. Higdon, C. Nakhleh, and B. Williams, “Cosmic calibration: Constraints from the matter power spectrum and the cosmic microwave background,” *Phys. Rev. D - Part. Fields, Gravit. Cosmol.*, vol. 76, no. 8, 2007, doi: 10.1103/PhysRevD.76.083503.

[110] T. Mui, R. Hu, and G. Zhang, “Uncertainty quantification on SAM simulations of EBR-II loss-of-flow tests,” *18th Int. Top. Meet. Nucl. React. Therm. Hydraul. NURETH 2019*, pp. 6217–6229, 2019.

[111] T. Sumner, T. Y. C. Wei, and A. Mohamed, “Benchmark Specifications and Data Requirements for EBR-II Shutdown Heat Removal Tests SHRT-17 and SHRT-45R EBR-II Benchmark Presentation Outline,” 2012.

[112] F. Chollet, “Keras,” *GitHub Repos.*, 2015, [Online]. Available: https://github.com/fchollet/keras.

[113] 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.*, vol. 206, no. 1, 2020, doi: 10.1080/00295450.2019.1626170.

[114] M. Zarei, R. Ghaderi, and A. Minuchehr, “Space independent xenon oscillations control in VVER reactor: A bifurcation analysis approach,” *Prog. Nucl. Energy*, vol. 88, pp. 19–27, 2016, doi: 10.1016/j.pnucene.2015.11.018.

[115] D. H. Bailey and P. N. Swarztrauber, “A Fast Method for the Numerical Evaluation of Continuous Fourier and Laplace Transforms,” *SIAM J. Sci. Comput.*, vol. 15, no. 5, 1994, Accessed: Nov. 12, 2018. [Online]. Available: http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/fourint.pdf.

[116] P. Bühlmann and S. van de Geer, “Non-convex loss functions and ℓ1-regularization,” in *Statistics for High-Dimensional Data*, Springer Berlin Heidelberg, 2011, pp. 293–338.

[117] L. E. O. Breiman and J. H. Friedman, “Estimating Optimal Transformations for Multiple Regression and Correlation,” *J. Am. Stat. Assoc.*, no. September, pp. 580–598, 1985.

[118] Nick Touran, “ace documentation — ace 0.2-1 documentation.” Accessed: Nov. 12, 2018. [Online]. Available: https://partofthething.com/ace/.

[119] “Deep Learning Toolbox - MATLAB.” Accessed: Nov. 12, 2018. [Online]. Available: https://www.mathworks.com/products/deep-learning.html.

[120] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, “On Dynamic Mode Decompostition: Theory and Applications,” no. Dmd, pp. 1–30.

[121] Y. Bang, H. S. Abdel-Khalik, and J. M. Hite, “Hybrid reduced order modeling applied to nonlinear models,” *Proc. 2011 Am. Control Conf.*, no. March, pp. 1885–1891, 2011, doi: 10.1002/nme.

[122] M. Verleysen and D. François, “The curse of dimensionality in data mining and time series prediction,” *Lect. Notes Comput. Sci.*, vol. 3512, pp. 758–770, 2005, doi: 10.1007/11494669_93.

[123] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Dynamic mode decomposition with control,” *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 1, pp. 142–161, 2014, doi: 10.1137/15M1013857.

[124] J. N. Kutz, “Data-Driven Modeling & Scientific Computation,” 2019.

[125] A. Lokhov, “Load-following with nuclear power plants,” no. 29, pp. 18–20, 2011.

[126] Nuclear Engineering Agency, “Technical and Economic Aspects of Load Following with Nuclear Power Plants,” 2011.

[127] M. P. Peterson and C. E. Paulsen, “An Implicit Steady-State Initialization Package for the RELAP5 Computer Code,” 1995.

[128] *RELAP5 / MOD3 . 3 CODE MANUAL VOLUME III : DEVELOPMENTAL ASSESSMENT*, vol. III. 2006.